



Panduan Pengguna

AWS IoT SiteWise



AWS IoT SiteWise: Panduan Pengguna

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS IoT SiteWise?	1
Bagaimana cara AWS IoT SiteWise kerja	2
Menelan data industri	2
Model aset untuk mengontekstualisasikan data yang dikumpulkan	3
Analisis menggunakan kueri, alarm, dan prediksi	4
Visualisasikan operasi	4
Menyimpan data	5
Mengintegrasikan dengan layanan lainnya	5
Gunakan kasus untuk AWS IoT SiteWise	5
Manufaktur	5
Makanan dan minuman	6
Energi dan utilitas	6
Bekerja dengan AWS SDKs	6
Konsep	7
Memulai	14
Persyaratan	14
Siapkan AWS akun	15
Mendaftar untuk Akun AWS	15
Buat pengguna dengan akses administratif	15
Gunakan demo mulai cepat	18
Buat AWS IoT SiteWise demo	18
Hapus AWS IoT SiteWise demo	20
Tutorial	22
Hitung OEE	22
Prasyarat	22
Cara menghitung OEE	23
Menelan data	25
Prasyarat	26
Langkah 1: Buat AWS IoT kebijakan	27
Langkah 2: Buat AWS IoT sesuatu	29
Langkah 3: Buat model aset perangkat	31
Langkah 4: Buat model aset armada perangkat	33
Langkah 5: Buat dan konfigurasikan aset perangkat	34
Langkah 6: Buat dan konfigurasikan aset armada perangkat	35

Langkah 7: Buat aturan di AWS IoT Core untuk mengirim data ke aset perangkat	36
Langkah 8: Jalankan skrip klien perangkat	39
Langkah 9: Bersihkan sumber daya setelah tutorial	47
Visualisasikan dan bagikan data di Monitor SiteWise	48
Prasyarat	49
Langkah 1: Buat portal di SiteWise Monitor	50
Langkah 2: Masuk ke portal	54
Langkah 3: Buat proyek ladang angin	56
Langkah 4: Buat dasbor untuk memvisualisasikan data ladang angin	60
Langkah 5: Jelajahi portal	67
Langkah 6: Bersihkan sumber daya setelah tutorial	68
Publikasikan ke Amazon DynamoDB	71
Prasyarat	71
Langkah 1: Konfigurasikan AWS IoT SiteWise untuk mempublikasikan pembaruan nilai properti	72
Langkah 2: Buat aturan di AWS IoT Core	74
Langkah 3: Buat tabel DynamoDB	76
Langkah 4: Konfigurasikan tindakan aturan DynamoDB	78
Langkah 5: Jelajahi data di DynamoDB	79
Langkah 6: Bersihkan sumber daya setelah tutorial	80
Menelan data ke AWS IoT SiteWise	84
Kelola aliran data	84
Konfigurasikan izin dan pengaturan	86
Kaitkan aliran data ke properti aset	87
Memutuskan aliran data dari properti aset	88
Menghapus aliran data	89
Memperbarui alias properti aset	90
Skenario umum	91
Menelan data dengan AWS IoT SiteWise APIs	93
BatchPutAssetPropertyValue API	93
CreateBulkImportJob API	96
Gunakan AWS IoT Core aturan	105
Berikan akses yang diperlukan	105
Konfigurasikan tindakan aturan	106
Kurangi biaya dengan Basic Ingest	115
Gunakan AWS IoT Events tindakan	116

Gunakan pengelola AWS IoT Greengrass aliran	117
Gunakan SiteWise gateway Edge	118
Konsep kunci gateway	118
Manfaat mengimplementasikan SiteWise Edge	119
Menghosting sendiri gateway	120
Persyaratan	121
Buat gateway	126
Instal perangkat lunak gateway	129
Gateway V3 berkemampuan MQTT	132
Aliran klasik, gateway V2	188
Tambahkan sumber data	202
Komponen untuk SiteWise Edge	245
Filter aset	247
Dukungan proxy dan toko kepercayaan	248
Gunakan APIs	254
Host gateway di Siemens Industrial Edge	272
Keamanan	273
Siemens Secure Storagedan aplikasi AWS IoT SiteWise Edge	273
Tujuan untuk Siemens Industrial Edge perangkat	274
Migrasi dari aplikasi pratinjau	275
Pemecahan Masalah	276
AWS IoT SiteWise Changelog aplikasi tepi	276
Persyaratan	278
Buat gateway	279
Buat Siemens Databus user	280
Akses aplikasi	281
Instal aplikasi	281
Perbarui konfigurasi aplikasi yang diinstal	284
Tujuan dan filter jalur	284
Memahami tujuan	285
Memahami filter jalur	288
Menambahkan tujuan waktu nyata	292
Tambahkan tujuan buffer menggunakan Amazon S3	300
Tambahkan filter jalur	307
Kelola tujuan	310
Kelola gateway	317

Kelola gateway SiteWise Edge Anda dengan AWS IoT SiteWise konsol	317
Kelola gateway SiteWise Edge menggunakan for AWS OpsHubAWS IoT SiteWise	318
Akses gateway SiteWise Edge Anda menggunakan kredensyal sistem operasi lokal	320
Kelola sertifikat gateway SiteWise Edge	322
Ubah versi paket komponen gateway SiteWise Edge	323
Daftar SiteWise gateway Edge	323
Jelaskan gateway SiteWise Edge	324
Buat gateway SiteWise Edge	325
Perbarui gateway SiteWise Edge	326
Perbarui konfigurasi kemampuan gateway	327
Tag sumber daya gateway	328
Tag daftar untuk gateway	329
Hapus tag dari gateway	330
Perbarui versi AWS IoT SiteWise komponen	331
Hapus gateway SiteWise Edge	332
Cadangkan dan pulihkan gateway	333
Backup harian data metrik	334
Kembalikan gateway SiteWise Edge	334
Kembalikan AWS IoT SiteWise data	335
Validasi backup dan restorasi yang berhasil	336
Gerbang warisan ()AWS IoT Greengrass Version 1	337
Model aset industri	339
Status aset dan model	341
Periksa status aset	341
Memeriksa status aset atau model komponen	343
Versi model aset	345
Mengambil versi aktif dari model aset atau model komponen (konsol)	346
Mengambil versi aktif dari model aset atau model komponen ()AWS CLI	347
Model komposit khusus (komponen)	348
Model komposit kustom sebaris	348
Component-model-based model komposit kustom	350
Gunakan jalur untuk mereferensikan properti model komposit khusus	352
Mengatur objek IDs	354
Bekerja dengan objek UUIDs	354
Gunakan eksternal IDs	355
Buat model	356

Buat model aset di AWS IoT SiteWise	357
Buat model komponen	372
Tentukan properti data	376
Buat model komposit khusus (komponen)	457
Buat aset	462
Buat aset (konsol)	462
Buat aset (AWS CLI)	463
Konfigurasi aset baru	464
Cari aset	465
Prasyarat	465
Pencarian lanjutan di Konsol AWS IoT SiteWise	465
Perbarui nilai atribut	468
Mengasosiasikan dan memisahkan aset	471
Mengaitkan dan memisahkan aset (konsol)	472
Mengasosiasikan dan memisahkan aset (AWS CLI)	473
Perbarui aset dan model	475
Perbarui aset di AWS IoT SiteWise	475
Perbarui model aset dan model komponen	477
Perbarui model komposit khusus (komponen)	482
Penguncian optimis untuk penulisan model aset	486
Hapus aset dan model di AWS IoT SiteWise	489
Hapus aset	490
Hapus model aset	492
Operasi massal dengan aset dan model	494
Konsep dan terminologi kunci	495
Fungsionalitas yang didukung	496
Prasyarat operasi massal	496
Jalankan pekerjaan impor massal	499
Jalankan pekerjaan ekspor massal	501
Pelacakan kemajuan pekerjaan dan penanganan kesalahan	505
Impor contoh metadata	510
Contoh metadata ekspor	525
Skema pekerjaan transfer metadata	528
Pantau data dengan alarm	547
Jenis alarm	547
Status alarm	549

Properti status alarm	549
Tentukan alarm pada model aset	552
Persyaratan untuk pemberitahuan alarm	556
Tentukan AWS IoT Events alarm	556
Tentukan alarm eksternal	591
Konfigurasikan alarm pada aset	593
Konfigurasikan nilai ambang batas (konsol)	594
Konfigurasikan nilai ambang (AWS CLI)	594
Konfigurasikan pengaturan notifikasi	597
Menanggapi alarm	599
Menanggapi alarm (konsol)	600
Menanggapi alarm (API)	603
Menelan status alarm eksternal	604
Memetakan aliran status alarm eksternal	604
Menelan data status alarm	606
AWS IoT SiteWise Asisten	608
Konfigurasikan AWS IoT SiteWise Asisten	608
Buat kumpulan data	610
Mengedit kumpulan data	615
Menghapus kumpulan data	617
AWS IoT SiteWise Pertanyaan asisten	618
Memantau data dengan AWS IoT SiteWise Monitor	619
SiteWise Memantau peran	620
Federasi SAFL	622
SiteWise Pantau konsep	623
Memulai dengan AWS IoT SiteWise Monitor (Klasik)	625
Buat portal	626
Konfigurasikan portal Anda	627
Mengundang administrator	631
Tambahkan pengguna portal	634
Buat dasbor (CLI)	638
Nyalakan alarm untuk portal Anda	644
Aktifkan portal Anda di tepi	647
Kelola portal Anda	647
Memulai dengan AWS IoT SiteWise Monitor (Ai-aware)	657
Buat portal	659

Konfigurasi portal Anda	659
Kelola portal Anda	662
Hapus portal	666
Buat dasbor dengan AWS CLI	667
Login portal	672
Membuat proyek	672
Perbarui proyek	673
Hapus proyek	674
Membuat sebuah Dasbor	674
Perbarui dasbor	676
Hapus dasbor	676
Konfigurasi dasbor	677
Data kueri dari AWS IoT SiteWise	698
Kueri nilai aset saat ini	699
Kueri nilai saat ini properti aset (konsol)	699
Kueri nilai saat ini properti aset (AWS CLI)	699
Kueri nilai properti aset historis	700
Agregat properti aset kueri	702
Agregat untuk properti aset (API)	703
Agregat untuk properti aset (AWS CLI)	704
AWS IoT SiteWise bahasa query	705
Prasyarat	706
Referensi bahasa kueri	707
Berinteraksi dengan layanan lain	715
Memahami properti aset dalam topik MQTT	716
Bekerja dengan notifikasi	716
Aktifkan notifikasi properti aset (konsol)	717
Aktifkan notifikasi properti aset (AWS CLI)	717
Pemberitahuan kueri	719
Ekspor data ke Amazon S3	722
Integrasi Grafana	722
Integrasi dengan AWS IoT TwinMaker	724
Mengaktifkan integrasi	725
Integrasi dan AWS IoT SiteWise AWS IoT TwinMaker	725
Mendeteksi anomali peralatan	726
Tambahkan definisi prediksi (konsol)	728

Latih prediksi (konsol)	731
Memulai atau menghentikan inferensi pada prediksi (konsol)	732
Tambahkan definisi prediksi (CLI)	733
Latih prediksi dan inferensi awal (CLI)	736
Melatih prediksi (CLI)	737
Memulai atau menghentikan inferensi pada prediksi (CLI)	739
Kelola penyimpanan data	742
Konfigurasi pengaturan penyimpanan	743
Dampak retensi data	743
Konfigurasi untuk tingkat hangat (konsol)	744
Konfigurasi untuk tingkat hangat (AWS CLI) (.....	745
Konfigurasi untuk tingkat dingin (konsol)	749
Konfigurasi untuk tingkat dingin (AWS CLI)	752
Memecahkan masalah pengaturan penyimpanan	757
Kesalahan: Bucket tidak ada	757
Kesalahan: Akses ditolak ke jalur Amazon S3	757
Kesalahan: Peran ARN tidak dapat diasumsikan	758
Kesalahan: Gagal mengakses bucket Amazon S3 lintas wilayah	758
Jalur file dan skema data yang disimpan di tingkat dingin	759
Data peralatan (pengukuran)	759
Metrik, transformasi, dan agregat	764
Metadata aset	768
Metadata hierarki aset	772
File indeks data penyimpanan	775
Contoh kode	776
Hal-hal mendasar	780
Halo AWS IoT SiteWise	781
Pelajari dasar-dasarnya	784
Tindakan	848
Keamanan	923
Perlindungan data	924
Privasi lalu lintas antarjaringan	925
AWS IoT SiteWise Peningkatan Layanan Asisten Bisnis	925
Enkripsi data	925
Enkripsi diam	926
Enkripsi bergerak	929

Manajemen kunci	930
Manajemen identitas dan akses	932
Audiens	933
Mengautentikasi menggunakan identitas	933
Bagaimana AWS IoT SiteWise bekerja dengan IAM	937
Kebijakan terkelola	956
Peran terkait layanan	960
Mengatur izin untuk alarm	978
Cross-service membingungkan deputi pencegahan di AWS IoT SiteWise	984
Memecahkan masalah identitas dan akses	985
Validasi kepatuhan	987
Ketahanan	988
Keamanan infrastruktur	989
Konfigurasi dan analisis kerentanan	990
Titik akhir VPC	991
Operasi API yang didukung	991
Membuat titik akhir VPC antarmuka	994
Akses AWS IoT SiteWise melalui titik akhir VPC antarmuka	994
Buat kebijakan titik akhir VPC	996
Praktik terbaik keamanan	997
Gunakan kredensi otentikasi di server OPC UA Anda	997
Gunakan mode komunikasi terenkripsi untuk server OPC UA Anda	997
Perbarui komponen Anda	997
Enkripsi sistem file gateway SiteWise Edge Anda	998
Akses aman ke konfigurasi tepi Anda	998
Mengamankan data Siemens Industrial Edge Management	998
Grant SiteWise Monitor pengguna izin minimum yang mungkin	998
Jangan mengekspos informasi sensitif	999
Ikuti praktik terbaik AWS IoT Greengrass keamanan	999
Lihat juga	999
Log dan memantau	1000
Memantau log layanan	1000
Mengelola masuk AWS IoT SiteWise	1002
Contoh: entri file AWS IoT SiteWise log	1004
Pantau log gateway SiteWise Edge	1004
Gunakan CloudWatch Log Amazon	1004

Gunakan log layanan	1006
Pantau dengan CloudWatch metrik Amazon	1008
AWS IoT Greengrass Version 2 metrik gateway	1008
Log panggilan API dengan AWS CloudTrail	1017
AWS IoT SiteWise informasi di CloudTrail	1018
AWS IoT SiteWise peristiwa data di CloudTrail	1019
AWS IoT SiteWise acara manajemen di CloudTrail	1021
Contoh: entri file AWS IoT SiteWise log	1021
Tandai sumber daya Anda	1024
Gunakan tag di AWS IoT SiteWise	1024
Tag dengan AWS Management Console	1024
Tag dengan AWS IoT SiteWise API	1025
Gunakan tag dengan kebijakan IAM	1026
Pemecahan Masalah	1028
Memecahkan masalah gateway	1028
Konfigurasi dan akses log gateway SiteWise Edge	1028
Memecahkan masalah gateway SiteWise Edge	1029
Memecahkan masalah aplikasi AWS IoT SiteWise Edge pada Siemens Industrial Edge	1035
Memecahkan masalah integrasi sumber terbuka di Edge	1036
Memecahkan masalah AWS IoT Greengrass	1039
Memecahkan masalah portal	1039
Pengguna dan administrator tidak dapat mengakses portal AWS IoT SiteWise	1039
Memecahkan masalah tindakan aturan AWS IoT SiteWise	1040
Konfigurasi AWS IoT Core log	1041
Konfigurasi tindakan kesalahan penerbitan ulang	1042
Memecahkan masalah aturan	1044
Memecahkan masalah aturan ()AWS IoT SiteWise	1046
Memecahkan masalah aturan (DynamoDB)	1048
Memecahkan masalah impor dan ekspor massal	1051
Titik akhir dan kuota	1052
Titik akhir	1052
Kuota	1052
AWS IoT SiteWise Batas pelambatan API asisten	1070
Kuota untuk deteksi anomali	1070
Riwayat dokumen	1071
.....	mxci

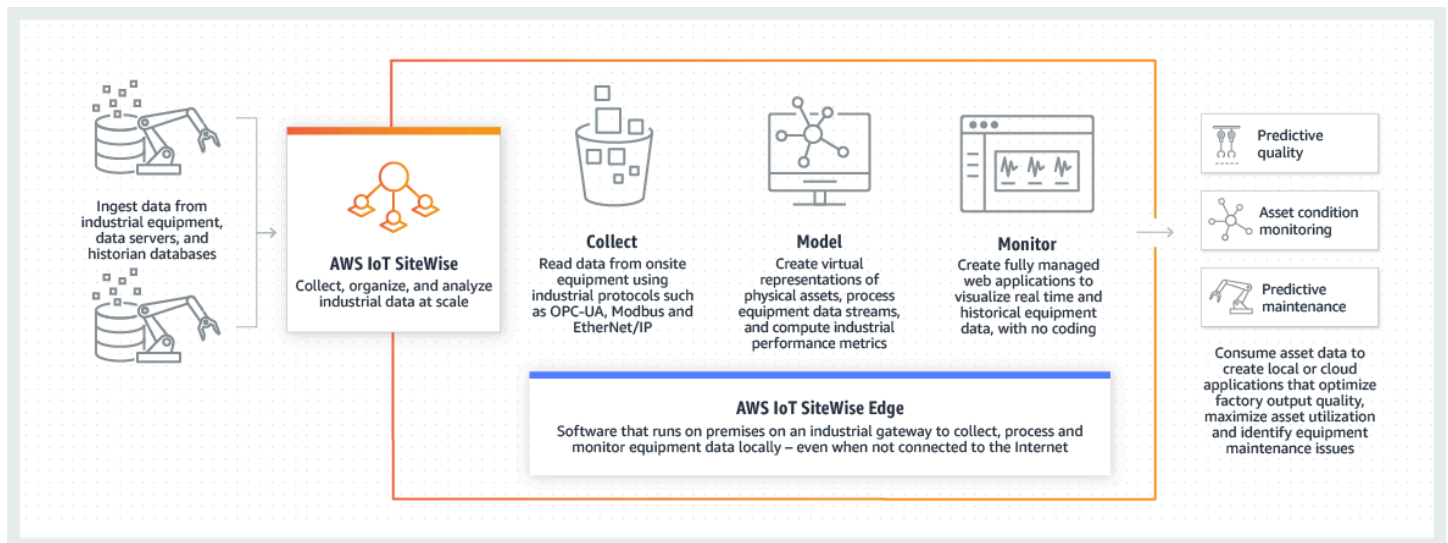
Apa itu AWS IoT SiteWise?

AWS IoT SiteWise adalah layanan terkelola yang dengannya Anda dapat mengumpulkan, menyimpan, mengatur, dan memantau data dari peralatan industri dalam skala besar untuk membantu Anda membuat keputusan yang lebih baik dan berbasis data. Anda dapat menggunakannya AWS IoT SiteWise untuk memantau operasi di seluruh fasilitas, dengan cepat menghitung metrik kinerja industri umum, dan membuat aplikasi yang menganalisis data peralatan industri untuk mencegah masalah peralatan yang mahal dan mengurangi kesenjangan dalam produksi.

Dengan AWS IoT SiteWise Monitor, pengguna operasional Anda dapat membuat aplikasi web untuk melihat dan menganalisis data industri Anda secara real-time. Anda dapat memperoleh wawasan tentang operasi industri Anda dengan mengonfigurasi dan memantau metrik seperti waktu rata-rata antara kegagalan dan efektivitas peralatan secara keseluruhan (OEE).

AWS IoT SiteWise Edge adalah komponen AWS IoT SiteWise yang memungkinkan pengumpulan, penyimpanan, dan pemrosesan data pada perangkat lokal. Ini berguna jika Anda memiliki akses terbatas ke internet atau perlu menjaga data Anda tetap pribadi.

Diagram berikut menunjukkan arsitektur dasar AWS IoT SiteWise:



Topik

- [Bagaimana cara AWS IoT SiteWise kerja](#)
- [Gunakan kasus untuk AWS IoT SiteWise](#)
- [Menggunakan layanan ini dengan AWS SDK](#)

- [AWS IoT SiteWise konsep](#)

Bagaimana cara AWS IoT SiteWise kerja

AWS IoT SiteWise menawarkan kerangka pemodelan sumber daya yang dapat Anda gunakan untuk membuat representasi perangkat, proses, dan fasilitas industri Anda. Representasi peralatan dan proses Anda disebut model aset di AWS IoT SiteWise. Dengan model aset, Anda menentukan data mentah yang akan dikonsumsi dan cara memprosesnya menjadi metrik yang berguna. Bangun dan visualisasikan aset dan model untuk operasi industri Anda di [AWS IoT SiteWise konsol](#). Anda juga dapat mengonfigurasi model aset untuk mengumpulkan dan memproses data di edge atau di AWS Cloud.

Topik

- [Menelan data industri](#)
- [Model aset untuk mengontekstualisasikan data yang dikumpulkan](#)
- [Analisis menggunakan kueri, alarm, dan prediksi](#)
- [Visualisasikan operasi](#)
- [Menyimpan data](#)
- [Mengintegrasikan dengan layanan lainnya](#)

Menelan data industri

Mulailah menggunakan AWS IoT SiteWise dengan menelan data industri. Menelan data Anda dilakukan dengan salah satu dari beberapa cara:

- Konsumsi langsung dari server di tempat: Gunakan protokol seperti OPC UA untuk membaca data langsung dari perangkat di tempat. Menyebarkan perangkat lunak gateway SiteWise Edge AWS IoT Greengrass V2, kompatibel dengan, pada berbagai platform seperti gateway industri umum atau server virtual. Anda dapat menghubungkan hingga 100 server OPC UA ke satu AWS IoT SiteWise gateway. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise Persyaratan gateway yang dihosting sendiri di Edge](#).

Perhatikan bahwa protokol seperti Modbus TCP dan Ethernet/IP (EIP) didukung melalui kemitraan kami dalam konteks. Domatica AWS IoT Greengrass V2

- Pemrosesan data tepi dengan paket: Tingkatkan gateway SiteWise Edge Anda dengan menambahkan paket untuk mengaktifkan kemampuan edge yang komprehensif. Dengan SiteWise

Edge, tersedia di AWS IoT Greengrass V2, pemrosesan data dijalankan langsung di tempat sebelum dikirim dengan aman ke AWS Cloud menggunakan AWS IoT Greengrass aliran. Untuk informasi selengkapnya, lihat [Siapkan sumber OPC UA di SiteWise Edge](#).

- Penyerapan adaptif melalui Amazon S3 dengan operasi massal: Saat bekerja dengan sejumlah besar aset atau model aset, gunakan operasi massal untuk mengimpor dan mengeksport sumber daya massal dari bucket Amazon S3. Untuk informasi selengkapnya, lihat [Operasi massal dengan aset dan model](#).
- Pesan MQTT dengan Aturan AWS IoT Inti: Untuk perangkat yang terhubung ke AWS IoT Core yang mengirim pesan MQTT, gunakan mesin aturan AWS IoT Inti untuk mengarahkan pesan tersebut ke AWS IoT SiteWise. Jika Anda memiliki perangkat yang terhubung ke AWS IoT Core yang mengirim pesan [MQTT](#), gunakan mesin aturan Inti untuk mengarahkan pesan tersebut. AWS IoT SiteWise Untuk informasi selengkapnya, lihat [Menelan data untuk AWS IoT SiteWise menggunakan aturan AWS IoT Core](#).
- Penyerapan data yang dipicu peristiwa: Gunakan AWS IoT Events tindakan untuk mengonfigurasi SiteWise tindakan IoT AWS IoT Events untuk mengirim data saat peristiwa terjadi. AWS IoT SiteWise Untuk informasi selengkapnya, lihat [Menelan data ke dari AWS IoT SiteWise AWS IoT Events](#).
- AWS IoT SiteWise API: Aplikasi Anda di Edge atau di cloud dapat langsung mengirim data ke AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Menelan data dengan AWS IoT SiteWise APIs](#).

Model aset untuk mengontekstualisasikan data yang dikumpulkan

Setelah menelan data, Anda dapat menggunakan data untuk membuat representasi virtual dari aset, proses, dan fasilitas Anda dengan membangun model operasi fisik Anda. Aset, yang mewakili perangkat atau proses, mentransmisikan aliran data ke Cloud. AWS Aset juga dapat menandakan pengelompokan perangkat logis. Hirarki dibentuk dengan mengaitkan aset untuk mencerminkan operasi yang kompleks. Hirarki ini memungkinkan aset untuk mengakses data dari aset anak terkait. Aset dibuat dari model aset. Model aset adalah struktur deklaratif yang menstandarisasi format aset. Gunakan kembali komponen aset untuk organisasi dan pemeliharaan model Anda. Untuk informasi selengkapnya, lihat [Model aset industri](#).

Dengan AWS IoT SiteWise, Anda dapat mengonfigurasi aset Anda untuk mengubah data yang masuk menjadi metrik dan transformasi kontekstual.

- Mengubah pekerjaan saat menerima data peralatan.
- Metrik dihitung pada interval yang Anda tentukan.

Metrik dan transformasi berlaku untuk aset individu atau beberapa aset. AWS IoT SiteWise secara otomatis menghitung agregat statistik yang umum digunakan seperti rata-rata, jumlah, dan hitungan, di berbagai kerangka waktu yang relevan dengan data, metrik, dan transformasi peralatan Anda.

Aset dapat disinkronkan menggunakan AWS IoT TwinMaker. Untuk informasi selengkapnya, lihat [Integrasi dan AWS IoT SiteWise AWS IoT TwinMaker](#).

Analisis menggunakan kueri, alarm, dan prediksi

Analisis tanggal yang dikumpulkan AWS IoT SiteWise dengan menjalankan kueri dan mengatur alarm. Anda juga dapat menggunakan Amazon Lookout untuk secara otomatis mendeteksi anomali dalam metrik dan mengidentifikasi akar penyebabnya.

- Atur alarm khusus untuk mengingatkan tim Anda saat peralatan atau proses menyimpang dari kinerja optimal, memastikan identifikasi dan penyelesaian masalah yang cepat. Untuk informasi selengkapnya, lihat [Pantau data dengan alarm di AWS IoT SiteWise](#).
- Gunakan operasi AWS IoT SiteWise API untuk menanyakan nilai saat ini, nilai historis, dan agregat properti aset Anda selama interval waktu tertentu. Untuk informasi selengkapnya, lihat [Data kueri dari AWS IoT SiteWise](#).
- Gunakan deteksi anomali dengan Amazon Lookout for Equipment untuk mengidentifikasi dan memvisualisasikan perubahan peralatan atau kondisi pengoperasian. Dengan deteksi anomali, Anda dapat menentukan langkah-langkah pemeliharaan preventif untuk operasi Anda. Integrasi ini memungkinkan pelanggan untuk menyinkronkan data antara AWS IoT SiteWise dan Amazon Lookout for Equipment. Untuk informasi selengkapnya, lihat [Deteksi anomali dengan Lookout for Equipment](#).

Visualisasikan operasi

Siapkan SiteWise Monitor untuk membuat aplikasi web untuk karyawan operasional Anda. Aplikasi web membantu karyawan untuk memvisualisasikan operasi Anda. Menangani berbagai tingkat akses untuk karyawan Anda menggunakan IAM Identity Center atau IAM. Konfigurasi login dan izin unik untuk setiap karyawan untuk melihat himpunan bagian tertentu dari seluruh operasi industri. AWS IoT SiteWise menyediakan [panduan aplikasi](#) bagi karyawan ini untuk mempelajari cara menggunakan SiteWise Monitor.

Untuk informasi selengkapnya tentang memvisualisasikan operasi Anda, lihat [Memantau data dengan AWS IoT SiteWise Monitor](#).

Menyimpan data

Anda dapat mengintegrasikan penyimpanan deret waktu dengan danau data industri Anda. AWS IoT SiteWise memiliki tiga tingkatan penyimpanan untuk data industri:

- Tingkat penyimpanan panas yang dioptimalkan untuk aplikasi waktu nyata.
- Tingkat penyimpanan hangat yang dioptimalkan untuk beban kerja analitis.
- Tingkat penyimpanan dingin yang dikelola pelanggan menggunakan Amazon S3 untuk aplikasi data operasional dengan toleransi latensi tinggi.

AWS IoT SiteWise membantu Anda mengelola biaya penyimpanan dengan menyimpan data terbaru di tingkat penyimpanan panas. Kemudian, Anda menentukan kebijakan penyimpanan data untuk memindahkan data historis ke penyimpanan tingkat hangat atau dingin. Untuk informasi selengkapnya, lihat [Mengelola penyimpanan data di AWS IoT SiteWise](#).

Anda juga dapat mengimpor dan mengekspor metadata aset. Untuk mengetahui informasi selengkapnya, lihat [Metadata aset](#).

Mengintegrasikan dengan layanan lainnya

AWS IoT SiteWise terintegrasi dengan beberapa AWS layanan untuk mengembangkan AWS IoT solusi lengkap di AWS Cloud. Untuk informasi selengkapnya, lihat [Berinteraksi dengan AWS layanan lain](#).

Gunakan kasus untuk AWS IoT SiteWise

AWS IoT SiteWise digunakan di berbagai industri untuk banyak pengumpulan data industri dan aplikasi analisis.

Kumpulkan data secara konsisten dari semua sumber Anda untuk membantu menyelesaikan masalah dengan cepat. AWS IoT SiteWise menawarkan pemantauan jarak jauh untuk mengumpulkan data secara langsung di tempat atau mengumpulkannya dari berbagai sumber di banyak fasilitas. AWS IoT SiteWise memberikan fleksibilitas yang diperlukan untuk solusi data IoT industri.

Manufaktur

AWS IoT SiteWise dapat menyederhanakan proses pengumpulan dan pemanfaatan data dari peralatan Anda untuk menentukan dan meminimalkan inefisiensi, meningkatkan operasi industri.

AWS IoT SiteWise membantu Anda mengumpulkan data dari jalur manufaktur dan peralatan. Dengan AWS IoT SiteWise, Anda dapat mentransfer data ke AWS Cloud dan membuat metrik kinerja untuk peralatan dan proses spesifik Anda. Anda dapat menggunakan metrik yang dihasilkan untuk memahami efektivitas keseluruhan operasi Anda dan mengidentifikasi peluang untuk inovasi dan peningkatan. Anda juga dapat melihat proses manufaktur Anda dan mengidentifikasi kekurangan peralatan dan proses, kesenjangan produksi, atau cacat produk.

Makanan dan minuman

Fasilitas industri makanan dan minuman menangani berbagai macam pengolahan makanan, termasuk menggiling biji-bijian menjadi tepung, menyembelih dan mengemas daging, dan merakit, memasak, dan membekukan makanan microwave. Pabrik pengolahan makanan sering menjangkau beberapa lokasi dengan operator pabrik dan peralatan di lokasi terpusat untuk memantau proses dan peralatan. Misalnya, unit pendingin menilai penanganan dan kedaluwarsa bahan. Mereka memantau pembuatan limbah di seluruh fasilitas untuk memastikan efisiensi operasional. Dengan AWS IoT SiteWise, Anda dapat mengelompokkan aliran data sensor dari beberapa lokasi berdasarkan jalur produksi, dan fasilitas sehingga teknisi proses Anda dapat lebih memahami dan melakukan perbaikan di seluruh fasilitas.

Energi dan utilitas

Dengan AWS IoT SiteWise, Anda dapat menyelesaikan masalah peralatan dengan lebih mudah dan lebih efisien. Anda dapat memantau kinerja aset dari jarak jauh dan secara real time. Akses data peralatan historis dari mana saja untuk menentukan potensi masalah, mengirimkan sumber daya yang akurat, dan mencegah dan memperbaiki masalah dengan lebih cepat.

Menggunakan layanan ini dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode

Dokumentasi SDK	Contoh kode
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode
Alat AWS untuk PowerShell	Alat AWS untuk PowerShell contoh kode
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

AWS IoT SiteWise konsep

Berikut ini adalah konsep inti dari AWS IoT SiteWise:

Agregat

Agregat adalah metrik fundamental, atau pengukuran, yang AWS IoT SiteWise secara otomatis menghitung untuk semua data deret waktu. Untuk informasi selengkapnya, lihat [Agregat properti aset kueri di AWS IoT SiteWise](#).

Aset

Saat Anda memasukkan, atau menelan, data AWS IoT SiteWise dari peralatan industri Anda, perangkat, peralatan, dan proses Anda masing-masing ditampilkan sebagai aset. Setiap aset memiliki data terkait. Misalnya, peralatan mungkin memiliki nomor seri, lokasi, merek dan model, dan tanggal pemasangan. Mungkin juga memiliki nilai deret waktu untuk ketersediaan, kinerja, kualitas, suhu, tekanan, dan banyak lagi. Kelompokkan aset ke dalam hierarki, memungkinkan aset untuk mengakses data yang disimpan dalam aset anak mereka. Untuk informasi selengkapnya, lihat [Model aset industri](#).

Hirarki aset

Siapkan hierarki aset untuk membuat representasi logis dari operasi industri Anda. Untuk melakukan ini, tentukan hierarki dalam model aset dan kaitkan aset yang dibuat dari model itu dengan hierarki yang ditentukan. Metrik dalam aset induk dapat menggabungkan data dari properti aset anak, memungkinkan Anda menghitung metrik yang menawarkan wawasan tentang keseluruhan operasi Anda atau bagian tertentu darinya. Untuk informasi selengkapnya, lihat [Tentukan hierarki model aset](#).

Model aset

Setiap aset dibuat menggunakan model aset. Model aset adalah struktur yang menentukan dan menstandarisasi format aset Anda. Mereka memastikan informasi yang konsisten di beberapa aset dengan jenis yang sama, memungkinkan Anda menangani data dalam aset yang mewakili grup perangkat. [Di setiap model aset, Anda dapat menentukan atribut, input deret waktu \(pengukuran\), transformasi deret waktu \(transformasi\), agregasi deret waktu \(metrik\), dan hierarki aset](#). Untuk informasi selengkapnya, lihat [Model aset industri](#).

Tentukan di mana properti model aset Anda diproses dengan mengonfigurasi model aset Anda untuk edge. Manfaatkan fitur ini untuk menangani dan memantau data aset di perangkat lokal Anda.

Properti aset

Properti aset adalah struktur dalam setiap aset yang menyimpan data industri. Setiap properti memiliki tipe data dan juga dapat memiliki unit. Properti dapat berupa [atribut](#), [pengukuran](#), [transformasi](#), atau [metrik](#). Untuk informasi selengkapnya, lihat [Tentukan properti data](#).

Konfigurasi properti aset untuk menghitung di tepi. Untuk informasi selengkapnya tentang pemrosesan data di edge, lihat [Siapkan sumber OPC UA di SiteWise Edge](#).

Atribut

Atribut adalah properti aset yang biasanya tetap konstan, seperti produsen perangkat atau lokasi perangkat. Atribut dapat memiliki nilai preset. Setiap aset yang dibuat dari model aset mencakup nilai default dari atribut yang ditentukan dalam model tersebut. Untuk informasi selengkapnya, lihat [Tentukan data statis \(atribut\)](#).

Dasbor

Setiap proyek berisi satu set dasbor. Dasbor menyediakan satu set visualisasi untuk nilai-nilai dari satu set aset. Pemilik proyek membuat dasbor dan visualisasi yang dikandungnya. Ketika pemilik proyek siap untuk membagikan kumpulan dasbor, pemilik dapat mengundang pemirsa ke proyek, yang memberi mereka akses ke semua dasbor dalam proyek. Jika Anda menginginkan kumpulan pemirsa yang berbeda untuk dasbor yang berbeda, Anda harus membagi dasbor di antara proyek. Saat pemirsa melihat dasbor, mereka dapat menyesuaikan rentang waktu untuk melihat data tertentu.

Dataset

Dataset adalah kumpulan data yang mewakili data deret waktu, data, dan non-time-series data non-peralatan seperti jadwal shift, catatan pemeliharaan, dan database karyawan. Mereka mendukung data eksternal dan menggunakan kemampuan AWS IoT SiteWise analitik. Ini termasuk sumber dataset, skema dataset dan parameter dataset. AWS IoT SiteWise Asisten menggunakan kumpulan data yang menggunakan indeks Amazon Kendra.

Aliran data

Masukan, atau konsumsi, data industri AWS IoT SiteWise bahkan sebelum membuat model aset dan aset. AWS IoT SiteWise secara otomatis menghasilkan aliran data untuk mengumpulkan aliran data mentah dari peralatan Anda.

Alias aliran data

Alias aliran data membantu Anda mengidentifikasi aliran data dengan mudah. Misalnya, alias `server1-windfarm/3/turbine/7/temperature` menunjukkan nilai suhu yang berasal

dari turbin #7 di ladang angin #3. Istilah `server1` ini adalah nama sumber data yang membantu mengidentifikasi server OPC UA, dan `server1-` merupakan awalan yang dilampirkan ke semua aliran data yang dilaporkan dari server OPC UA ini.

Asosiasi aliran data

Setelah Anda membuat model aset dan aset, kaitkan aliran data dengan properti aset yang ditentukan dalam aset Anda untuk menyusun data Anda. AWS IoT SiteWise kemudian dapat menggunakan model aset dan aset untuk menangani data yang masuk dari aliran data Anda. Anda juga dapat memisahkan aliran data dari properti aset. Untuk informasi selengkapnya, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Destinasi

Tujuan di SiteWise Edge mewakili titik akhir tempat Anda ingin mengirim telemetri atau data yang diproses. SiteWise Edge mendukung AWS IoT SiteWise hot tier, buffered ingestion, atau bucket Amazon S3 sebagai tujuan. Anda dapat mengonfigurasi tujuan untuk berlangganan topik MQTT tertentu menggunakan filter jalur. Untuk informasi selengkapnya, lihat [Memahami tujuan AWS IoT SiteWise Edge](#).

Formula

Setiap properti [transformasi](#) dan [metrik](#) dilengkapi dengan rumus yang menguraikan bagaimana properti mengubah atau mengumpulkan data. Rumus ini mencakup input properti, operator, dan fungsi yang ditawarkan oleh. AWS IoT SiteWise Untuk informasi selengkapnya, lihat [Gunakan ekspresi rumus](#).

Pengukuran

Pengukuran adalah properti aset yang menggambarkan aliran data deret waktu sensor mentah dari perangkat atau peralatan. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).

Metrik

Metrik adalah properti aset yang mewakili data deret waktu agregat. Setiap metrik disertai dengan ekspresi matematika ([rumus](#)) yang menguraikan cara mengumpulkan titik data dan interval waktu untuk menghitung agregasi tersebut. Metrik menghasilkan satu titik data untuk setiap interval waktu yang ditentukan. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).

MQTT

MQTT (Message Queuing Telemetry Transport) adalah protokol pesan ringan untuk sensor dan perangkat.

Paket

SiteWise Edge gateway menggunakan paket untuk menentukan cara mengumpulkan, memproses, dan merutekan data. Untuk informasi selengkapnya tentang paket yang tersedia untuk gateway SiteWise Edge Anda, lihat [the section called “Gunakan paket”](#).

Paket pengumpulan data

Gunakan paket pengumpulan data sehingga gateway SiteWise Edge Anda dapat mengumpulkan data industri Anda dan mengarahkannya ke AWS tujuan pilihan Anda.

Paket pengolahan data

Gunakan paket pemrosesan data untuk memproses, menyimpan, dan mengambil data Anda di tepi hingga 30 hari. Tukar data yang diproses tepi ke dan dari aplikasi lokal melalui SiteWise Edge. APIs

OPC

OPC UA (Open Platform Communications Unified Architecture) adalah protokol komunikasi untuk otomasi industri.

Filter jalur

Gunakan filter jalur dalam gateway untuk berlangganan topik MQTT dan mempublikasikan ke AWS IoT SiteWise tujuan yang didukung. Sumber berbasis MQTT, jalur pemrosesan data, dan tujuan semuanya bertukar data menggunakan topik MQTT pada gateway V3 berkemampuan MQTT yang dihosting sendiri. Anda dapat menentukan filter topik untuk menentukan data yang ingin Anda konsumsi atau rute ke tujuan yang berbeda.

Portal

AWS IoT SiteWise Monitor Portal adalah aplikasi web yang dapat Anda gunakan untuk memvisualisasikan dan berbagi AWS IoT SiteWise data Anda. Portal memiliki satu atau lebih administrator dan berisi nol atau lebih proyek.

Administrator portal

Setiap portal SiteWise Monitor memiliki satu atau lebih administrator portal. Administrator portal menggunakan portal untuk membuat proyek yang berisi koleksi aset dan dasbor. Administrator portal kemudian menetapkan aset dan pemilik untuk setiap proyek. Dengan mengontrol akses ke

proyek, administrator portal menentukan aset mana yang dapat dilihat oleh pemilik proyek dan pemirsa.

Proyek

Setiap portal SiteWise Monitor berisi serangkaian proyek. Setiap proyek memiliki subset AWS IoT SiteWise aset Anda yang terkait dengannya. Pemilik proyek membuat satu atau beberapa dasbor untuk menyediakan cara yang konsisten untuk melihat data yang terkait dengan aset tersebut. Pemilik proyek dapat mengundang pemirsa ke proyek untuk memungkinkan mereka melihat aset dan dasbor dalam proyek. Proyek ini adalah unit dasar berbagi dalam SiteWise Monitor. Pemilik proyek dapat mengundang pengguna yang diberi akses ke portal oleh AWS administrator. Pengguna harus memiliki akses ke portal sebelum proyek di portal itu dapat dibagikan dengan pengguna tersebut.

Pemilik proyek

Setiap proyek SiteWise Monitor memiliki pemilik. Pemilik proyek membuat visualisasi dalam bentuk dasbor untuk mewakili data operasional secara konsisten. Ketika dasbor siap untuk dibagikan, pemilik proyek dapat mengundang pemirsa ke proyek. Pemilik proyek juga dapat menugaskan pemilik lain ke proyek. Pemilik proyek dapat mengonfigurasi ambang batas dan pengaturan notifikasi untuk alarm.

Penampil proyek

Setiap proyek SiteWise Monitor memiliki pemirsa. Pemirsa proyek dapat terhubung ke portal untuk melihat dasbor yang dibuat oleh pemilik proyek. Di setiap dasbor, pemirsa proyek dapat menyesuaikan rentang waktu untuk lebih memahami data operasional. Pemirsa proyek hanya dapat melihat dasbor di proyek yang dapat mereka akses. Pemirsa proyek dapat mengenali dan menunda alarm.

Alias properti

Anda memiliki opsi untuk membuat alias pada properti aset, seperti jalur aliran data server OPC UA (misalnya, /company/windfarm/3/turbine/7/temperature), menyederhanakan identifikasi properti aset selama konsumsi atau pengambilan data aset. Saat Anda menggunakan [gateway SiteWise Edge](#) untuk mengambil data dari server, alias properti Anda harus cocok dengan jalur aliran data mentah Anda. Untuk informasi selengkapnya, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Pemberitahuan properti

Saat Anda mengaktifkan pemberitahuan properti untuk properti aset, AWS IoT SiteWise menerbitkan pesan MQTT ke AWS IoT Core setiap kali properti tersebut menerima nilai baru.

Payload pesan mencakup detail tentang pembaruan ke nilai properti tersebut. Gunakan notifikasi nilai properti untuk membuat solusi yang menghubungkan data industri Anda AWS IoT SiteWise dengan AWS layanan lain. Untuk informasi selengkapnya, lihat [Berinteraksi dengan AWS layanan lain](#).

SiteWise Gerbang tepi

Gateway SiteWise Edge dipasang di tempat pelanggan untuk mengumpulkan, menangani, dan mengarahkan data. Gateway SiteWise Edge terhubung ke sumber data industri Anda melalui berbagai protokol untuk mengumpulkan dan memproses data, mengirimkannya ke cloud. AWS SiteWise Edge gateway juga dapat terhubung ke sumber data mitra. Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise gateway Edge](#).

Transformasi

Transformasi adalah properti aset yang mewakili data deret waktu yang ditransformasikan. Setiap transformasi disertai dengan ekspresi matematika ([rumus](#)) yang menentukan bagaimana mengkonversi titik data dari satu bentuk ke bentuk lainnya. Titik data yang ditransformasikan memiliki one-to-one hubungan dengan titik data input. Untuk informasi selengkapnya, lihat [Mengubah data \(mengubah\)](#).

Visualisasi

Di setiap dasbor, pemilik proyek memutuskan bagaimana menampilkan properti dan alarm aset yang terkait dengan proyek. Ketersediaan dapat direpresentasikan sebagai grafik garis, sementara nilai lainnya mungkin ditampilkan sebagai grafik batang atau indikator kinerja utama (KPIs). Alarm paling baik ditampilkan sebagai kisi-kisi status dan jadwal status. Pemilik proyek menyesuaikan setiap visualisasi untuk memberikan pemahaman terbaik tentang data untuk aset tersebut.

Memulai dengan AWS IoT SiteWise

Dengan AWS IoT SiteWise, Anda dapat mengumpulkan, mengatur, menganalisis, dan memvisualisasikan data Anda.

AWS IoT SiteWise menyediakan demo yang dapat Anda gunakan untuk menjelajahi layanan tanpa mengonfigurasi sumber data nyata. Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise demo](#).

Anda dapat menyelesaikan tutorial berikut untuk menjelajahi fitur-fitur tertentu dari AWS IoT SiteWise:

- [Menelan data AWS IoT SiteWise dari AWS IoT berbagai hal](#)
- [Visualisasikan dan bagikan data ladang angin di Monitor SiteWise](#)
- [Publikasikan pembaruan nilai properti ke Amazon DynamoDB](#)

Lihat topik berikut untuk mempelajari lebih lanjut tentang AWS IoT SiteWise:

- [Menelan data ke AWS IoT SiteWise](#)
- [Model aset industri](#)
- [Konfigurasi kemampuan tepi di AWS IoT SiteWise Edge](#)
- [Memantau data dengan AWS IoT SiteWise Monitor](#)
- [Data kueri dari AWS IoT SiteWise](#)
- [Berinteraksi dengan AWS layanan lain](#)

Topik

- [Persyaratan](#)
- [Siapkan AWS akun](#)

Persyaratan

Anda harus memiliki AWS akun untuk memulai AWS IoT SiteWise. Jika Anda tidak memilikinya, lihat bagian berikut untuk petunjuk tentang cara mengatur akun.

Gunakan Wilayah di mana AWS IoT SiteWise tersedia. Untuk informasi lebih lanjut, lihat [AWS IoT SiteWise kuota dan titik akhir](#). Anda dapat menggunakan pemilih Wilayah di AWS Management Console untuk beralih ke salah satu Wilayah ini.

Siapkan AWS akun

Topik

- [Mendaftar untuk Akun AWS](#)
- [Buat pengguna dengan akses administratif](#)

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/pendaftaran>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS mengirimkan Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk <https://aws.amazon.com/ke/> dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

Untuk bantuan masuk dengan menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) di AWS Sign-In Panduan Pengguna.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root \(konsol\) Anda](#) di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal AWS akses](#) di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

Untuk petunjuknya, lihat [Membuat set izin](#) di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

Untuk petunjuk, lihat [Menambahkan grup](#) di Panduan AWS IAM Identity Center Pengguna.

Gunakan AWS IoT SiteWise demo

Anda dapat dengan mudah menjelajahi AWS IoT SiteWise dengan menggunakan AWS IoT SiteWise demo. AWS IoT SiteWise menyediakan demo sebagai AWS CloudFormation template yang dapat Anda terapkan untuk membuat model aset, aset, dan portal SiteWise Monitor, dan menghasilkan data sampel hingga satu minggu.

Important

Setelah Anda membuat demo, Anda akan mulai dikenakan biaya untuk sumber daya yang dibuat dan dikonsumsi demo ini.

Topik

- [Buat AWS IoT SiteWise demo](#)
- [Hapus AWS IoT SiteWise demo](#)

Buat AWS IoT SiteWise demo

Anda dapat membuat AWS IoT SiteWise demo dari AWS IoT SiteWise konsol.

Note

Demo membuat fungsi Lambda, satu aturan CloudWatch Acara, dan peran AWS Identity and Access Management (IAM) yang diperlukan untuk demo. Anda mungkin melihat sumber daya ini di AWS akun Anda. Kami menyarankan Anda menyimpan sumber daya ini sampai Anda selesai dengan demo. Jika Anda menghapus sumber daya, demo mungkin berhenti bekerja dengan benar.

Untuk membuat demo di AWS IoT SiteWise konsol

1. Arahkan ke [AWS IoT SiteWise konsol](#) dan temukan SiteWise demo di sudut kanan atas halaman.
2. (Opsional) Di bawah SiteWise demo, ubah bidang Days to keep demo assets untuk menentukan berapa hari untuk menyimpan demo sebelum menghapusnya.

3. (Opsional) Untuk membuat portal SiteWise Monitor untuk memantau data sampel, lakukan hal berikut.

 Note

Anda akan dikenakan biaya untuk sumber daya SiteWise Monitor yang dibuat dan dikonsumsi oleh demo ini. Untuk informasi selengkapnya, lihat [SiteWise Monitor](#) di AWS IoT SiteWise Harga.

- a. Pilih Monitor Sumber Daya.
- b. Pilih Izin.
- c. Pilih peran IAM yang ada yang memberi pengguna IAM federasi Anda akses ke portal.

 Important

Peran IAM Anda harus memiliki izin berikut.

```
{
  "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": "*"
    }
  ]
}
```

Untuk informasi selengkapnya tentang cara bekerja dengan SiteWise Monitor, lihat [Apa itu AWS IoT SiteWise Monitor?](#) dalam Panduan AWS IoT SiteWise Monitor Aplikasi.

4. Pilih Buat demo.

Demo membutuhkan waktu sekitar 3 menit untuk dibuat. Jika demo gagal dibuat, akun Anda mungkin memiliki izin yang tidak mencukupi. Beralih ke akun yang memiliki izin administratif, atau gunakan langkah-langkah berikut untuk menghapus demo dan coba lagi:

a. Pilih Hapus demo.

Demo membutuhkan waktu sekitar 15 menit untuk dihapus.

b. Jika demo tidak dihapus, buka [AWS CloudFormation konsol](#), pilih tumpukan bernama IoT Site WiseDemoAssets, dan pilih Hapus di sudut kanan atas.

c. Jika demo gagal dihapus lagi, ikuti langkah-langkah di AWS CloudFormation konsol untuk melewati sumber daya yang gagal dihapus, dan coba lagi.

5. Setelah demo berhasil dibuat, Anda dapat menjelajahi aset dan data demo di [AWS IoT SiteWise konsol](#).

Hapus AWS IoT SiteWise demo

AWS IoT SiteWise Demo menghapus dirinya sendiri setelah seminggu, atau jumlah hari yang Anda pilih jika Anda membuat tumpukan demo dari AWS CloudFormation konsol. Anda dapat menghapus demo sebelumnya jika Anda selesai menggunakan sumber daya demo. Anda juga dapat menghapus demo jika demo gagal dibuat. Gunakan langkah-langkah berikut untuk menghapus demo secara manual.

Untuk menghapus AWS IoT SiteWise demo

1. Navigasikan ke [konsol AWS CloudFormation](#) tersebut.
2. Pilih IoT SiteWise Demo Assets dari daftar Stacks.
3. Pilih Hapus.

Saat Anda menghapus tumpukan, semua sumber daya yang dibuat untuk demo akan dihapus.

4. Dalam dialog konfirmasi, pilih Hapus tumpukan.

Tumpukan membutuhkan waktu sekitar 15 menit untuk dihapus. Jika demo gagal dihapus, pilih Hapus di sudut kanan atas lagi. Jika demo gagal dihapus lagi, ikuti langkah-langkah di AWS CloudFormation konsol untuk melewati sumber daya yang gagal dihapus, dan coba lagi.

AWS IoT SiteWise tutorial

Selamat datang di halaman AWS IoT SiteWise tutorial. Kumpulan tutorial yang berkembang ini memberdayakan Anda dengan pengetahuan dan keterampilan yang dibutuhkan untuk menavigasi seluk-beluk. AWS IoT SiteWise Tutorial ini menawarkan beragam topik dasar untuk memenuhi kebutuhan Anda. Saat Anda mempelajari tutorial, temukan wawasan yang tak ternilai ke dalam berbagai aspek. AWS IoT SiteWise

Setiap tutorial menggunakan contoh peralatan tertentu. Tutorial ini ditujukan untuk lingkungan pengujian, dan mereka menggunakan nama perusahaan fiktif, model, aset, properti, dan sebagainya. Tujuannya adalah untuk memberikan pedoman umum. Tutorial tidak dimaksudkan untuk penggunaan langsung dalam lingkungan produksi tanpa tinjauan dan adaptasi yang cermat untuk memenuhi kebutuhan unik organisasi Anda.

Topik

- [Hitung OEE di AWS IoT SiteWise](#)
- [Menelan data AWS IoT SiteWise dari AWS IoT berbagai hal](#)
- [Visualisasikan dan bagikan data ladang angin di Monitor SiteWise](#)
- [Publikasikan pembaruan nilai properti ke Amazon DynamoDB](#)

Hitung OEE di AWS IoT SiteWise

Tutorial ini memberikan contoh bagaimana menghitung efektivitas peralatan secara keseluruhan (OEE) untuk proses manufaktur. Akibatnya, perhitungan atau rumus OEE Anda mungkin berbeda dari yang ditunjukkan di sini. Secara umum, OEE didefinisikan sebagai $\text{Availability} * \text{Quality} * \text{Performance}$. Untuk mempelajari lebih lanjut tentang menghitung OEE, lihat [Efektivitas peralatan secara keseluruhan](#) di Wikipedia.

Prasyarat

Untuk menyelesaikan tutorial ini, Anda harus mengonfigurasi konsumsi data untuk perangkat yang memiliki tiga aliran data berikut:

- `Equipment_State`— Kode numerik yang mewakili keadaan mesin, seperti idle, kesalahan, penghentian yang direncanakan, atau operasi normal.

- **Good_Count**— Aliran data di mana setiap titik data berisi jumlah operasi yang berhasil sejak titik data terakhir.
- **Bad_Count**— Aliran data di mana setiap titik data berisi jumlah operasi yang gagal sejak titik data terakhir.

Untuk mengonfigurasi konsumsi data, lihat. [Menelan data ke AWS IoT SiteWise](#) Jika Anda tidak memiliki operasi industri yang tersedia, Anda dapat menulis skrip yang menghasilkan dan mengunggah data sampel melalui AWS IoT SiteWise API.

Cara menghitung OEE

Dalam tutorial ini, Anda membuat model aset yang menghitung OEE dari tiga aliran input data: **Equipment_State**, **Good_Count** dan **Bad_Count**. Dalam contoh ini, pertimbangkan mesin pengemas generik, seperti yang digunakan untuk mengemas gula, keripik kentang, atau cat. Di [AWS IoT SiteWise konsol](#), buat model AWS IoT SiteWise aset dengan pengukuran, transformasi, dan metrik berikut. Kemudian, Anda dapat membuat aset untuk mewakili mesin pengemas dan mengamati bagaimana AWS IoT SiteWise menghitung OEE.

Tentukan [pengukuran](#) berikut untuk mewakili aliran data mentah dari mesin pengemas.

Pengukuran

- **Equipment_State**— Aliran data (atau pengukuran) yang menyediakan keadaan mesin pengemas saat ini dalam kode numerik:
 - 1024— Mesin menganggur.
 - 1020— Kesalahan, seperti kesalahan atau penundaan.
 - 1000Sebuah perhentian yang direncanakan.
 - 1111Operasi normal.
- **Good_Count**— Aliran data di mana setiap titik data berisi jumlah operasi yang berhasil sejak titik data terakhir.
- **Bad_Count**— Aliran data di mana setiap titik data berisi jumlah operasi yang gagal sejak titik data terakhir.

Dengan menggunakan aliran data **Equipment_State** pengukuran dan kode yang dikandungnya, tentukan [transformasi](#) berikut (atau pengukuran turunan). Transformasi memiliki one-to-one hubungan dengan pengukuran mentah.

Transformasi

- `Idle = eq(Equipment_State, 1024)`— Aliran data yang diubah yang berisi keadaan idle mesin.
- `Fault = eq(Equipment_State, 1020)`— Aliran data yang diubah yang berisi status kesalahan mesin.
- `Stop = eq(Equipment_State, 1000)`— Aliran data yang diubah yang berisi status berhenti yang direncanakan mesin.
- `Running = eq(Equipment_State, 1111)`— Aliran data yang diubah yang berisi keadaan operasional normal mesin.

Dengan menggunakan pengukuran mentah dan pengukuran yang diubah, tentukan [metrik](#) berikut yang menggabungkan data mesin selama interval waktu yang ditentukan. Pilih interval waktu yang sama untuk setiap metrik saat Anda menentukan metrik di bagian ini.

Metrik

- `Successes = sum(Good_Count)`— Jumlah paket yang berhasil diisi selama interval waktu yang ditentukan.
- `Failures = sum(Bad_Count)`— Jumlah paket yang tidak berhasil diisi selama interval waktu yang ditentukan.
- `Idle_Time = statetime(Idle)`— Total waktu idle mesin (dalam detik) per interval waktu yang ditentukan.
- `Fault_Time = statetime(Fault)`— Total waktu kesalahan mesin (dalam detik) per interval waktu yang ditentukan.
- `Stop_Time = statetime(Stop)`— Total waktu berhenti yang direncanakan mesin (dalam detik) per interval waktu yang ditentukan.
- `Run_Time = statetime(Running)`— Total waktu mesin (dalam detik) berjalan tanpa masalah per interval waktu yang ditentukan.
- `Down_Time = Idle_Time + Fault_Time + Stop_Time`— Total downtime mesin (dalam detik) selama interval waktu yang ditentukan, dihitung sebagai jumlah status mesin selain `Run_Time`.
- `Availability = Run_Time / (Run_Time + Down_Time)`— Waktu kerja mesin atau persentase waktu terjadwal bahwa mesin tersedia untuk beroperasi selama interval waktu yang ditentukan.

- $Quality = Successes / (Successes + Failures)$ — Persentase mesin dari paket yang berhasil diisi selama interval waktu yang ditentukan.
- $Performance = ((Successes + Failures) / Run\_Time) / Ideal\_Run\_Rate$ — Kinerja mesin selama interval waktu yang ditentukan sebagai persentase dari laju lari ideal (dalam detik) untuk proses Anda.

Misalnya, Anda `Ideal_Run_Rate` mungkin 60 paket per menit (1 paket per detik). Jika Anda `Ideal_Run_Rate` per menit atau per jam, Anda perlu membaginya dengan faktor konversi unit yang sesuai `Run_Time` karena dalam hitungan detik.

- $OEE = Availability * Quality * Performance$ — Efektivitas peralatan mesin secara keseluruhan selama interval waktu yang ditentukan. Rumus ini menghitung OEE sebagai pecahan dari 1.

Note

Jika OEE didefinisikan sebagai transformasi, nilai output dihitung untuk masing-masing nilai input. Ada potensi untuk menghasilkan nilai tak terduga karena evaluasi transformasi mempertimbangkan nilai terbaru yang tersedia untuk semua properti yang berkontribusi dalam rumus. Untuk pembaruan properti dengan stempel waktu yang sama, nilai keluaran dapat ditimpa oleh pembaruan dari properti masuk lainnya. Misalnya ketika Ketersediaan, Kualitas, dan Kinerja dihitung, OEE dihitung dengan titik data terakhir yang tersedia untuk dua properti lainnya. Nilai-nilai yang berkontribusi ini berbagi stempel waktu, dan menyebabkan nilai keluaran OEE yang salah. Pesanan tidak dijamin untuk mengubah perhitungan.

Menelan data AWS IoT SiteWise dari AWS IoT berbagai hal

Pelajari cara menyerap data AWS IoT SiteWise dari armada AWS IoT benda dengan menggunakan bayangan perangkat dalam tutorial ini. Bayangan perangkat adalah objek JSON yang menyimpan informasi status saat ini untuk AWS IoT perangkat. Untuk informasi selengkapnya, lihat [Layanan bayangan perangkat](#) di Panduan AWS IoT Pengembang.

Setelah Anda menyelesaikan tutorial ini, Anda dapat mengatur operasi AWS IoT SiteWise berdasarkan AWS IoT hal-hal. Dengan menggunakan AWS IoT berbagai hal, Anda dapat mengintegrasikan operasi Anda dengan fitur berguna lainnya dari AWS IoT. Misalnya, Anda dapat mengonfigurasi AWS IoT fitur untuk melakukan tugas-tugas berikut:

- Konfigurasi aturan tambahan untuk mengalirkan data ke [AWS IoT Events](#), [Amazon DynamoDB](#), dan layanan lainnya. AWS Untuk informasi selengkapnya, lihat [Aturan](#) di Panduan AWS IoT Pengembang.
- Indeks, telusuri, dan agregat data perangkat Anda dengan layanan pengindeksan AWS IoT armada. Untuk informasi selengkapnya, lihat [layanan pengindeksan armada](#) di AWS IoT Panduan Pengembang.
- Audit dan amankan perangkat Anda dengan AWS IoT Device Defender. Untuk informasi lebih lanjut, lihat [AWS IoT Device Defender](#) dalam Panduan Pengembang AWS IoT .

Dalam tutorial ini, Anda mempelajari cara menyalin data dari bayangan perangkat AWS IoT benda ke aset. AWS IoT SiteWise Untuk melakukannya, Anda membuat satu atau beberapa AWS IoT hal dan menjalankan skrip yang memperbarui bayangan perangkat setiap hal dengan data penggunaan CPU dan memori. Anda menggunakan data penggunaan CPU dan memori dalam tutorial ini untuk meniru data sensor yang realistis. Kemudian, Anda membuat aturan dengan AWS IoT SiteWise tindakan yang mengirimkan data ini ke aset AWS IoT SiteWise setiap kali bayangan perangkat diperbarui. Untuk informasi selengkapnya, lihat [Menyalin data untuk AWS IoT SiteWise menggunakan aturan AWS IoT Core](#).

Topik

- [Prasyarat](#)
- [Langkah 1: Buat AWS IoT kebijakan](#)
- [Langkah 2: Buat dan konfigurasi AWS IoT sesuatu](#)
- [Langkah 3: Buat model aset perangkat](#)
- [Langkah 4: Buat model aset armada perangkat](#)
- [Langkah 5: Buat dan konfigurasi aset perangkat](#)
- [Langkah 6: Buat dan konfigurasi aset armada perangkat](#)
- [Langkah 7: Buat aturan di AWS IoT Core untuk mengirim data ke aset perangkat](#)
- [Langkah 8: Jalankan skrip klien perangkat](#)
- [Langkah 9: Bersihkan sumber daya setelah tutorial](#)

Prasyarat

Untuk menyelesaikan tutorial ini, Anda memerlukan hal berikut:

- AWS Akun. Jika Anda tidak memilikinya, lihat [Siapkan AWS akun](#).
- Sebuah komputer pengembangan yang berjalan Windows macOS, Linux, atau Unix untuk mengakses AWS Management Console. Untuk informasi lebih lanjut, lihat [Memulai dengan AWS Management Console](#).
- Pengguna AWS Identity and Access Management (IAM) dengan izin administrator.
- Python3 diinstal pada komputer pengembangan Anda atau diinstal pada perangkat yang ingin Anda daftarkan sebagai AWS IoT sesuatu.

Langkah 1: Buat AWS IoT kebijakan

Dalam prosedur ini, buat AWS IoT kebijakan yang memungkinkan AWS IoT barang-barang Anda mengakses sumber daya yang digunakan dalam tutorial ini.

Untuk membuat AWS IoT kebijakan

1. Masuk ke [AWS Management Console](#).
2. Tinjau [AWS Wilayah](#) di AWS IoT SiteWise mana didukung. Beralih ke salah satu Wilayah yang didukung ini, jika perlu.
3. Navigasikan ke [konsol AWS IoT](#) tersebut. Jika tombol Connect device muncul, pilih tombol tersebut.
4. Di panel navigasi kiri, pilih Keamanan, lalu pilih Kebijakan.
5. Pilih Buat.
6. Masukkan nama untuk AWS IoT kebijakan (misalnya, **SiteWiseTutorialDevicePolicy**).
7. Di bawah Dokumen kebijakan, pilih JSON untuk memasukkan kebijakan berikut dalam formulir JSON. Ganti *region* dan *account-id* dengan Wilayah dan ID akun Anda, seperti **us-east-1** dan **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*"
}
]
```

Kebijakan ini memungkinkan AWS IoT perangkat Anda untuk membuat koneksi dan berkomunikasi dengan bayangan perangkat menggunakan pesan MQTT. Untuk informasi selengkapnya tentang pesan MQTT, lihat [Apa itu MQTT?](#) . Untuk berinteraksi dengan bayangan perangkat, AWS IoT barang-barang Anda mempublikasikan dan menerima pesan MQTT tentang topik yang dimulai. `$aws/things/thing-name/shadow/` Kebijakan ini menggabungkan variabel kebijakan sesuatu yang dikenal sebagai `${iot:Connection.Thing.ThingName}`. Variabel ini menggantikan nama benda yang terhubung di setiap topik. `iot:Connect` Pernyataan tersebut menetapkan batasan pada perangkat mana yang dapat membuat koneksi, memastikan bahwa variabel kebijakan hal hanya dapat menggantikan nama yang dimulai dengan `SiteWiseTutorialDevice`.

Untuk informasi selengkapnya, lihat [Variabel kebijakan Thing](#) di Panduan AWS IoT Pengembang.

Note

Kebijakan ini berlaku untuk hal-hal yang namanya dimulai `SiteWiseTutorialDevice`. Untuk menggunakan nama yang berbeda untuk barang-barang Anda, Anda harus memperbarui kebijakan yang sesuai.

8. Pilih Buat.

Langkah 2: Buat dan konfigurasi AWS IoT sesuatu

Dalam prosedur ini, Anda membuat dan mengkonfigurasi AWS IoT sesuatu. Anda dapat menunjuk komputer pengembangan Anda sebagai AWS IoT sesuatu. Saat Anda maju, ingatlah bahwa prinsip-prinsip yang Anda pelajari di sini dapat diterapkan pada proyek yang sebenarnya. Anda memiliki fleksibilitas untuk membuat dan mengatur AWS IoT hal-hal di perangkat apa pun yang mampu

menjalankan AWS IoT SDK, termasuk AWS IoT Greengrass dan FreeRTOS. Untuk informasi lebih lanjut, lihat [AWS IoT SDKs](#) dalam Panduan Pengembang AWS IoT .

Untuk membuat dan mengkonfigurasi AWS IoT sesuatu

1. Buka baris perintah dan jalankan perintah berikut untuk membuat direktori untuk tutorial ini.

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

2. Jalankan perintah berikut untuk membuat direktori untuk sertifikat barang Anda.

```
mkdir device1
```

Jika Anda membuat hal-hal tambahan, tambahkan nomor dalam nama direktori yang sesuai untuk melacak sertifikat mana yang termasuk dalam hal itu.

3. Navigasikan ke [konsol AWS IoT](#) tersebut.
4. Di panel navigasi kiri, pilih Semua perangkat di bagian Kelola. Lalu, pilih Berbagai hal.
5. Jika kotak dialog Anda belum memiliki hal apa pun muncul, pilih Buat sesuatu. Jika tidak, pilih Buat sesuatu.
6. Pada halaman Creating things, pilih Create a single dan kemudian pilih Next.
7. Pada halaman Tentukan properti benda, masukkan nama untuk AWS IoT barang Anda (misalnya, **SiteWiseTutorialDevice1**) lalu pilih Berikutnya. Jika Anda membuat hal-hal tambahan, tambahkan nomor dalam nama benda yang sesuai.

 Important

Nama benda harus sesuai dengan nama yang digunakan dalam kebijakan yang Anda buat di Langkah 1: Membuat AWS IoT kebijakan. Jika tidak, perangkat Anda tidak dapat terhubung AWS IoT.

8. Pada halaman Konfigurasi sertifikat perangkat - opsional, pilih Buat otomatis sertifikat baru (disarankan), lalu pilih Berikutnya. Sertifikat memungkinkan AWS IoT untuk mengidentifikasi perangkat Anda dengan aman.
9. Pada halaman Lampirkan kebijakan ke sertifikat - opsional, pilih kebijakan yang Anda buat di Langkah 1: Membuat AWS IoT kebijakan dan pilih Buat sesuatu.
10. Pada kotak dialog Unduh sertifikat dan kunci, lakukan hal berikut:

- a. Pilih tautan Unduh untuk mengunduh sertifikat barang Anda, kunci publik, dan kunci pribadi. Simpan ketiga file ke direktori yang Anda buat untuk sertifikat barang Anda (misalnya, `iot-sitewise-rule-tutorial/device1`).

 Important

Ini adalah satu-satunya saat Anda dapat mengunduh sertifikat dan kunci barang Anda, yang Anda perlukan agar perangkat Anda berhasil terhubung AWS IoT.

- b. Pilih tautan Unduh untuk mengunduh sertifikat CA root. Simpan sertifikat CA root ke file `iot-sitewise-rule-tutorial`. Kami merekomendasikan mengunduh Amazon Root CA 1.

11. Pilih Selesai.

Anda sekarang telah mendaftarkan AWS IoT sesuatu di komputer Anda. Ambil salah satu langkah berikut:

- Lanjutkan ke Langkah 3: Membuat model aset perangkat tanpa membuat AWS IoT hal-hal tambahan. Anda dapat menyelesaikan tutorial ini hanya dengan satu hal.
- Ulangi langkah-langkah di bagian ini di komputer atau perangkat lain untuk membuat lebih banyak AWS IoT hal. Untuk tutorial ini, kami sarankan Anda mengikuti opsi ini sehingga Anda dapat menelan data penggunaan CPU dan memori yang unik dari beberapa perangkat.
- Ulangi langkah-langkah di bagian ini pada perangkat yang sama (komputer Anda) untuk membuat lebih banyak AWS IoT hal. Setiap AWS IoT benda menerima data penggunaan CPU dan memori yang serupa dari komputer Anda, jadi gunakan pendekatan ini untuk menunjukkan pengambilan data non-unik dari beberapa perangkat.

Langkah 3: Buat model aset perangkat

Dalam prosedur ini, Anda membuat model aset AWS IoT SiteWise untuk mewakili perangkat Anda yang mengalirkan data penggunaan CPU dan memori. Untuk memproses data dalam aset yang mewakili kelompok perangkat, model aset menerapkan informasi yang konsisten di beberapa aset dengan jenis yang sama. Untuk informasi selengkapnya, lihat [Model aset industri](#).

Untuk membuat model aset yang mewakili perangkat

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di panel navigasi kiri, pilih Model.
3. Pilih Buat model.
4. Di bawah Detail model, masukkan nama untuk model Anda. Misalnya, **SiteWise Tutorial Device Model**.
5. Di bawah definisi Pengukuran, lakukan hal berikut:
 - a. Di Nama, masukkan **CPU Usage**.
 - b. Di Unit, masukkan%.
 - c. Biarkan tipe Data sebagai Ganda.

Properti pengukuran mewakili aliran data mentah perangkat. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).

6. Pilih Tambahkan pengukuran baru untuk menambahkan properti pengukuran kedua.
7. Di baris kedua di bawah Definisi pengukuran, lakukan hal berikut:
 - a. Di Nama, masukkan **Memory Usage**.
 - b. Di Unit, masukkan%.
 - c. Biarkan tipe Data sebagai Ganda.
8. Di bawah definisi Metrik, lakukan hal berikut:
 - a. Di Nama, masukkan **Average CPU Usage**.
 - b. Dalam Formula, masukkan **avg(CPU Usage)**. Pilih CPU Usage dari daftar pelengkapan otomatis saat muncul.
 - c. Dalam Interval waktu, masukkan **5 minutes**.

Properti metrik mendefinisikan perhitungan agregasi yang memproses semua titik data input selama interval dan mengeluarkan satu titik data per interval. Properti metrik ini menghitung rata-rata penggunaan CPU setiap perangkat setiap 5 menit. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).

9. Pilih Tambahkan metrik baru untuk menambahkan properti metrik kedua.
10. Di baris kedua di bawah Definisi metrik, lakukan hal berikut:
 - a. Di Nama, masukkan **Average Memory Usage**.

- b. Dalam Formula, masukkan **avg(Memory Usage)**. Pilih Memory Usage dari daftar pelengkapan otomatis saat muncul.
- c. Dalam Interval waktu, masukkan **5 minutes**.

Properti metrik ini menghitung penggunaan memori rata-rata setiap perangkat setiap 5 menit.

11. (Opsional) Tambahkan metrik tambahan lain yang ingin Anda hitung per perangkat. Beberapa fungsi menarik termasuk `min` dan `max`. Untuk informasi selengkapnya, lihat [Gunakan ekspresi rumus](#). Pada Langkah 4: Membuat model aset armada perangkat, Anda membuat aset induk yang dapat menghitung metrik menggunakan data dari seluruh armada perangkat Anda.
12. Pilih Buat model.

Langkah 4: Buat model aset armada perangkat

Dalam prosedur ini, Anda membuat model aset AWS IoT SiteWise untuk melambangkan koleksi perangkat Anda. Dalam model aset ini, Anda membuat struktur yang memungkinkan Anda menautkan banyak aset perangkat ke satu aset armada menyeluruh. Setelah itu, Anda menguraikan metrik dalam model aset armada untuk mengkonsolidasikan data dari semua aset perangkat yang terhubung. Pendekatan ini memberi Anda wawasan komprehensif tentang kinerja kolektif seluruh armada Anda.

Untuk membuat model aset yang mewakili armada perangkat

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Model.
3. Pilih Buat model.
4. Di bawah Detail model, masukkan nama untuk model Anda. Misalnya, **SiteWise Tutorial Device Fleet Model**.
5. Di bawah definisi Hierarki, lakukan hal berikut:
 - a. Dalam nama Hierarki, masukkan **Device**.
 - b. Dalam model Hierarki, pilih model aset perangkat Anda (**SiteWise Tutorial Device Model**).

Hierarki mendefinisikan hubungan antara model aset induk (armada) dan model aset anak (perangkat). Aset induk dapat mengakses data properti aset anak. Saat Anda membuat aset

nanti, Anda perlu mengaitkan aset anak ke aset induk sesuai dengan definisi hierarki dalam model aset induk. Untuk informasi selengkapnya, lihat [Tentukan hierarki model aset](#).

6. Di bawah definisi Metrik, lakukan hal berikut:
 - a. Di Nama, masukkan **Average CPU Usage**.
 - b. Dalam Formula, masukkan **avg(Device | Average CPU Usage)**. Ketika daftar pelengkapan otomatis muncul, pilih Device untuk memilih hierarki, lalu pilih Average CPU Usage untuk memilih metrik dari aset perangkat yang Anda buat sebelumnya.
 - c. Dalam Interval waktu, masukkan **5 minutes**.

Properti metrik ini menghitung penggunaan CPU rata-rata dari semua aset perangkat yang terkait dengan aset armada melalui **Device** hierarki.

7. Pilih Tambahkan metrik baru untuk menambahkan properti metrik kedua.
8. Di baris kedua di bawah Definisi metrik, lakukan hal berikut:
 - a. Di Nama, masukkan **Average Memory Usage**.
 - b. Dalam Formula, masukkan **avg(Device | Average Memory Usage)**. Ketika daftar pelengkapan otomatis muncul, pilih Device untuk memilih hierarki, lalu pilih Average Memory Usage untuk memilih metrik dari aset perangkat yang Anda buat sebelumnya.
 - c. Dalam Interval waktu, masukkan **5 minutes**.

Properti metrik ini menghitung penggunaan memori rata-rata dari semua aset perangkat yang terkait dengan aset armada melalui **Device** hierarki.

9. (Opsional) Tambahkan metrik tambahan lain yang ingin Anda hitung di seluruh armada perangkat Anda.
10. Pilih Buat model.

Langkah 5: Buat dan konfigurasi aset perangkat

Dalam prosedur ini, Anda menghasilkan aset perangkat yang didasarkan pada model aset perangkat Anda. Kemudian, Anda menentukan alias properti untuk setiap properti pengukuran. Alias properti adalah string unik yang mengidentifikasi properti aset. Kemudian, Anda dapat mengidentifikasi properti untuk upload data dengan menggunakan alias, bukan ID aset dan ID properti. Untuk informasi selengkapnya, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Untuk membuat aset perangkat dan menentukan alias properti

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Aset.
3. Pilih Buat aset.
4. Di bawah Informasi model, pilih model aset perangkat Anda **SiteWise Tutorial Device Model**.
5. Di bawah informasi Aset, masukkan nama untuk aset Anda. Misalnya, **SiteWise Tutorial Device 1**.
6. Pilih Buat aset.
7. Untuk aset perangkat baru Anda, pilih Edit.
8. Di bawah CPU Usage, masukkan **/tutorial/device/SiteWiseTutorialDevice1/cpu** sebagai alias properti. Anda menyertakan nama AWS IoT benda dalam alias properti, sehingga Anda dapat menyerap data dari semua perangkat Anda menggunakan satu AWS IoT aturan.
9. Di bawah Memory Usage, masukkan **/tutorial/device/SiteWiseTutorialDevice1/memory** sebagai alias properti.
10. Pilih Simpan.

Jika Anda membuat beberapa AWS IoT hal sebelumnya, ulangi langkah 3 hingga 10 untuk setiap perangkat, dan tambahkan nomor dalam nama aset dan alias properti yang sesuai. Misalnya, nama aset perangkat kedua harus **SiteWise Tutorial Device 2**, dan alias propertinya harus **/tutorial/device/SiteWiseTutorialDevice2/cpu**, dan **/tutorial/device/SiteWiseTutorialDevice2/memory**.

Langkah 6: Buat dan konfigurasi aset armada perangkat

Dalam prosedur ini, Anda membentuk aset armada perangkat yang berasal dari model aset armada perangkat Anda. Kemudian, Anda menautkan aset perangkat individual Anda ke aset armada. Asosiasi ini memungkinkan properti metrik aset armada untuk mengkompilasi dan menganalisis data dari beberapa perangkat. Data ini memberi Anda pandangan konsolidasi tentang kinerja kolektif seluruh armada.

Untuk membuat aset armada perangkat dan mengaitkan aset perangkat

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Aset.

3. Pilih Buat aset.
4. Di bawah Informasi model, pilih model aset armada perangkat Anda **SiteWise Tutorial Device Fleet Model**.
5. Di bawah informasi Aset, masukkan nama untuk aset Anda. Misalnya, **SiteWise Tutorial Device Fleet 1**.
6. Pilih Buat aset.
7. Untuk aset armada perangkat baru Anda, pilih Edit.
8. Di bawah Aset yang terkait dengan aset ini, pilih Tambahkan aset terkait dan lakukan hal berikut:
 - a. Di bawah Hierarki, pilih Device. Hierarki ini mengidentifikasi hubungan hierarkis antara perangkat dan aset armada perangkat. Anda mendefinisikan hierarki ini dalam model aset armada perangkat sebelumnya dalam tutorial ini.
 - b. Di bawah Aset, pilih aset perangkat Anda, SiteWise Tutorial Device1.
9. (Opsional) Jika Anda membuat beberapa aset perangkat lebih awal, ulangi langkah 8 hingga 10 untuk setiap aset perangkat yang Anda buat.
10. Pilih Simpan.

Anda sekarang akan melihat aset perangkat Anda diatur sebagai hierarki.

Langkah 7: Buat aturan di AWS IoT Core untuk mengirim data ke aset perangkat

Dalam prosedur ini, Anda membuat aturan di AWS IoT Core. Aturan ini dirancang untuk menafsirkan pesan notifikasi dari bayangan perangkat dan mengirimkan data ke aset perangkat Anda di AWS IoT SiteWise. Setiap kali bayangan perangkat Anda diperbarui, AWS IoT mengirimkan pesan MQTT. Anda dapat membuat aturan yang mengambil tindakan saat bayangan perangkat berubah berdasarkan pesan MQTT. Dalam hal ini, tujuannya adalah untuk menangani pesan pembaruan, mengekstrak nilai properti, dan mengirimkannya ke aset perangkat Anda AWS IoT SiteWise.

Untuk membuat aturan dengan AWS IoT SiteWise tindakan

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi kiri, pilih Perutean pesan, lalu pilih Aturan.
3. Pilih Buat aturan.
4. Masukkan nama dan deskripsi untuk aturan Anda dan pilih Berikutnya.

5. Masukkan pernyataan SQL berikut dan pilih Berikutnya.

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

Pernyataan kueri aturan ini berfungsi karena layanan bayangan perangkat menerbitkan pembaruan bayangan ke `$aws/things/thingName/shadow/update/accepted`. Untuk informasi selengkapnya tentang bayangan [perangkat, lihat Layanan bayangan perangkat](#) di Panduan AWS IoT Pengembang.

Dalam WHERE klausa, pernyataan kueri aturan ini menggunakan `topic(3)` fungsi untuk mendapatkan nama benda dari segmen ketiga topik. Kemudian, pernyataan tersebut menyaring perangkat yang memiliki nama yang tidak cocok dengan perangkat tutorial. Untuk informasi selengkapnya tentang AWS IoT SQL, lihat [referensi AWS IoT SQL](#) di Panduan AWS IoT Pengembang.

6. Di bawah Tindakan aturan, pilih Kirim data pesan ke properti aset AWS IoT SiteWise dan lakukan hal berikut:
- Pilih Berdasarkan alias properti.
 - Dalam alias Properti, masukkan **`/tutorial/device/${topic(3)}/cpu`**.

`${...}` Sintaks adalah template substitusi. AWS IoT mengevaluasi isi dalam kawat gigi. Template substitusi ini menarik nama benda dari topik untuk membuat alias unik untuk setiap hal. Untuk informasi selengkapnya, lihat [Templat substitusi](#) di Panduan AWS IoT Pengembang.

 Note

Karena ekspresi dalam templat substitusi dievaluasi secara terpisah dari SELECT pernyataan, Anda tidak dapat menggunakan templat substitusi untuk mereferensikan alias yang dibuat menggunakan klausa. AS Anda hanya dapat mereferensikan informasi yang ada di muatan asli, selain fungsi dan operator yang didukung.

- c. Di ID Entri - opsional, masukkan **`${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}`**.

Entri IDs secara unik mengidentifikasi setiap upaya entri nilai. Jika entri mengembalikan kesalahan, Anda dapat menemukan ID entri di output kesalahan untuk memecahkan masalah. Template substitusi dalam ID entri ini menggabungkan nama benda dan stempel waktu perangkat yang dilaporkan. Misalnya, ID entri yang dihasilkan mungkin terlihat seperti `SiteWiseTutorialDevice1-cpu-1579808494`.

- d. Dalam Waktu dalam hitungan detik, masukkan **`${floor(state.reported.timestamp)}`**.

Template substitusi ini menghitung waktu dalam hitungan detik dari stempel waktu yang dilaporkan perangkat. Dalam tutorial ini, perangkat melaporkan stempel waktu dalam hitungan detik dalam waktu epoch Unix sebagai nomor floating point.

- e. Di Offset dalam nanos - opsional, masukkan. **`${floor((state.reported.timestamp % 1) * 1E9)}`**

Template substitusi ini menghitung offset nanodetik dari waktu dalam hitungan detik dengan mengubah bagian desimal dari stempel waktu yang dilaporkan perangkat.

Note

AWS IoT SiteWise mengharuskan data Anda memiliki stempel waktu saat ini dalam waktu epoch Unix. Jika perangkat Anda tidak melaporkan waktu secara akurat, Anda bisa mendapatkan waktu saat ini dari mesin AWS IoT aturan dengan [stempel waktu \(\)](#). Fungsi ini melaporkan waktu dalam milidetik, jadi Anda harus memperbarui parameter waktu tindakan aturan Anda ke nilai berikut:

- Dalam Waktu dalam hitungan detik, masukkan **`${floor(timestamp() / 1E3)}`**.
- Di Offset di nanos, masukkan. **`${(timestamp() % 1E3) * 1E6}`**

- f. Di Tipe data, pilih Ganda.

Tipe data ini harus cocok dengan tipe data properti aset yang Anda tentukan dalam model aset.

- g. Dalam Value (Nilai), masukkan **`${state.reported.cpu}`**. Dalam template substitusi, Anda menggunakan `.` operator untuk mengambil nilai dari dalam struktur JSON.

- h. Pilih Tambah entri untuk menambahkan entri baru untuk properti penggunaan memori, dan selesaikan langkah-langkah berikut lagi untuk properti itu:
 - i. Pilih Berdasarkan alias properti.
 - ii. Dalam alias Properti, masukkan **/tutorial/device/\${topic(3)}/memory**.
 - iii. Di ID Entri - opsional, masukkan **\${concat(topic(3), "-memory-", floor(state.reported.timestamp))}**.
 - iv. Dalam Waktu dalam hitungan detik, masukkan **\${floor(state.reported.timestamp)}**.
 - v. Di Offset dalam nanos - opsional, masukkan **\${floor((state.reported.timestamp % 1) * 1E9)}**.
 - vi. Di Tipe data, pilih Ganda.
 - vii. Dalam Value (Nilai), masukkan **\${state.reported.memory}**.
 - i. Di bawah Peran IAM, pilih Buat peran baru untuk membuat peran IAM untuk tindakan aturan ini. Peran ini memungkinkan AWS IoT untuk mendorong data ke properti di aset armada perangkat Anda dan hierarki asetnya.
 - j. Masukkan nama peran dan pilih Buat.
7. (Opsional) Konfigurasi tindakan kesalahan yang dapat Anda gunakan untuk memecahkan masalah aturan Anda. Untuk informasi selengkapnya, lihat [Memecahkan masalah aturan \(\)AWS IoT SiteWise](#).
 8. Pilih Berikutnya.
 9. Tinjau pengaturan dan pilih Buat untuk membuat aturan.

Langkah 8: Jalankan skrip klien perangkat

Untuk tutorial ini, Anda tidak menggunakan perangkat yang sebenarnya untuk melaporkan data. Sebagai gantinya, Anda menjalankan skrip untuk memperbarui bayangan perangkat AWS IoT benda Anda dengan CPU dan penggunaan memori untuk meniru data sensor nyata. Untuk menjalankan skrip, Anda harus terlebih dahulu menginstal Python paket yang diperlukan. Dalam prosedur ini, Anda menginstal Python paket yang diperlukan dan kemudian menjalankan skrip klien perangkat.

Untuk mengkonfigurasi dan menjalankan skrip klien perangkat

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di bagian bawah panel navigasi kiri, pilih Pengaturan.

3. Simpan titik akhir kustom Anda untuk digunakan dengan skrip klien perangkat. Anda menggunakan titik akhir ini untuk berinteraksi dengan bayangan benda Anda. Titik akhir ini unik untuk akun Anda di Wilayah saat ini.

Titik akhir kustom Anda akan terlihat seperti contoh berikut.

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

4. Buka baris perintah dan jalankan perintah berikut untuk menavigasi ke direktori tutorial yang Anda buat sebelumnya.

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

5. Jalankan perintah berikut untuk menginstal file AWS IoT Device SDK for Python.

```
pip3 install AWSIoTPythonSDK
```

Untuk informasi selengkapnya, lihat [AWS IoT Device SDK for Python](#) di Panduan AWS IoT Pengembang

6. Jalankan perintah berikut untuk menginstal psutil, proses lintas platform dan perpustakaan utilitas sistem.

```
pip3 install psutil
```

Untuk informasi selengkapnya, lihat [psutil](#) di Python Package Index.

7. Buat file yang disebut `thing_performance.py` dalam `iot-sitewise-rule-tutorial` direktori dan kemudian salin kode Python berikut ke dalam file.

```
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, performance))
            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. Jalankan `thing_performance.py` dari baris perintah dengan parameter berikut:

- `-n, --thingName` - Nama benda Anda, seperti **SiteWiseTutorialDevice1**.
- `-e, --endpoint` — AWS IoT Titik akhir kustom Anda yang Anda simpan sebelumnya dalam prosedur ini.
- `-r, --rootCA` — Jalur ke sertifikat CA AWS IoT root Anda.

- -c, --cert — Jalan menuju sertifikat AWS IoT barang Anda.
- -k, --key — Jalan menuju kunci pribadi sertifikat AWS IoT barang Anda.
- -d, --requestDelay — (Opsional) Waktu dalam hitungan detik untuk menunggu di antara setiap pembaruan bayangan perangkat. Default ke 1 detik.
- -v, --enableLogging — (Opsional) Jika parameter ini ada, skrip mencetak pesan debug dari AWS IoT Device SDK for Python.

Perintah Anda akan terlihat mirip dengan contoh berikut.

```
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
```

Jika Anda menjalankan skrip untuk AWS IoT hal-hal tambahan, perbarui nama benda dan direktori sertifikat yang sesuai.

9. Coba buka dan tutup program di perangkat Anda untuk melihat bagaimana penggunaan CPU dan memori berubah. Skrip mencetak setiap pembacaan penggunaan CPU dan memori. Jika skrip mengunggah data ke layanan bayangan perangkat berhasil, output skrip akan terlihat seperti contoh berikut.

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

10. Ikuti langkah-langkah ini untuk memverifikasi bahwa skrip memperbarui bayangan perangkat:
 - a. Navigasikan ke [konsol AWS IoT](#) tersebut.
 - b. Di panel navigasi kiri, pilih Semua perangkat dan kemudian pilih Things.
 - c. Pilih barang Anda, SiteWiseTutorialDevice.
 - d. Pilih tab Device Shadows, pilih Classic Shadow, dan verifikasi bahwa status Shadow terlihat seperti contoh berikut.

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

Jika status bayangan benda Anda kosong atau tidak terlihat seperti contoh sebelumnya, periksa apakah skrip sedang berjalan dan berhasil terhubung AWS IoT. Jika skrip terus habis saat menghubungkan ke AWS IoT, periksa apakah [kebijakan hal](#) Anda dikonfigurasi sesuai dengan tutorial ini.

11. Ikuti langkah-langkah ini untuk memverifikasi bahwa tindakan aturan mengirim data ke AWS IoT SiteWise:
 - a. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
 - b. Di panel navigasi kiri, pilih Aset.
 - c. Pilih tanda panah di samping aset armada perangkat Anda (SiteWise Tutorial Device Fleet 11) untuk memperluas hierarki asetnya, lalu pilih aset perangkat Anda (SiteWise Tutorial Device1).
 - d. Pilih Pengukuran.
 - e. Verifikasi bahwa sel nilai terbaru memiliki nilai untuk CPU Usagedan Memory Usage properti.

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. Jika Memory Usage properti CPU Usagedan tidak memiliki nilai terbaru, segarkan halaman. Jika nilai tidak muncul setelah beberapa menit, lihat [Memecahkan masalah aturan \(\)AWS IoT SiteWise](#).

Anda telah menyelesaikan tutorial ini. Jika Anda ingin menjelajahi visualisasi langsung data Anda, Anda dapat mengonfigurasi portal di AWS IoT SiteWise Monitor Untuk informasi selengkapnya, lihat [Memantau data dengan AWS IoT SiteWise Monitor](#). Jika tidak, Anda dapat menekan CTRL+C di prompt perintah Anda untuk menghentikan skrip klien perangkat. Tidak mungkin program Python

akan mengirim pesan yang cukup untuk dikenakan biaya, tetapi ini adalah praktik terbaik untuk menghentikan program ketika Anda selesai.

Langkah 9: Bersihkan sumber daya setelah tutorial

Setelah Anda menyelesaikan tutorial tentang menelan data dari AWS IoT berbagai hal, bersihkan sumber daya Anda untuk menghindari biaya tambahan.

Untuk menghapus aset hierarkis di AWS IoT SiteWise

1. Arahkan ke [AWS IoT SiteWise konsol](#)
2. Di panel navigasi kiri, pilih Aset.
3. Saat Anda menghapus aset AWS IoT SiteWise, Anda harus terlebih dahulu memisahkannya.

Selesaikan langkah-langkah berikut untuk memisahkan aset perangkat Anda dari aset armada perangkat Anda:

- a. Pilih aset armada perangkat Anda (SiteWise Tutorial Device Fleet 1).
- b. Pilih Edit.
- c. Di bawah Aset yang terkait dengan aset ini, pilih Putuskan hubungan untuk setiap aset perangkat yang terkait dengan aset armada perangkat ini.
- d. Pilih Simpan.

Sekarang Anda akan melihat aset perangkat Anda tidak lagi terorganisir sebagai hierarki.

4. Pilih aset perangkat Anda (SiteWise Tutorial Device1).
5. Pilih Hapus.
6. Pada dialog konfirmasi, masukkan **Delete** lalu pilih Hapus.
7. Ulangi langkah 4 hingga 6 untuk setiap aset perangkat dan aset armada perangkat (SiteWise Tutorial Device Fleet 1).

Untuk menghapus model aset hierarkis di AWS IoT SiteWise

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Jika Anda belum melakukannya, hapus aset armada perangkat dan perangkat Anda. Untuk informasi lebih lanjut, lihat [prosedur sebelumnya](#). Anda tidak dapat menghapus model jika Anda memiliki aset yang dibuat dari model tersebut.

3. Di panel navigasi kiri, pilih Model.
4. Pilih model aset armada perangkat Anda (SiteWise Tutorial Device Fleet Model).

Saat menghapus model aset hierarkis, mulailah dengan menghapus model aset induk terlebih dahulu.

5. Pilih Hapus.
6. Pada dialog konfirmasi, masukkan **Delete** lalu pilih Hapus.
7. Ulangi langkah 4 hingga 6 untuk model aset perangkat Anda (SiteWise Tutorial Device Model).

Untuk menonaktifkan atau menghapus aturan di AWS IoT Core

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi kiri, pilih Perutean pesan, lalu pilih Aturan.
3. Pilih aturan Anda dan pilih Hapus.
4. Dalam dialog konfirmasi, masukkan nama aturan dan kemudian pilih Hapus.

Visualisasikan dan bagikan data ladang angin di Monitor SiteWise

Tutorial ini menjelaskan cara menggunakan AWS IoT SiteWise Monitor untuk memvisualisasikan dan berbagi data industri melalui aplikasi web yang dikelola, yang dikenal sebagai portal. Setiap portal mencakup proyek, memberi Anda fleksibilitas untuk memilih data mana yang dapat diakses dalam setiap proyek. Kemudian, tentukan orang-orang di organisasi Anda yang dapat mengakses setiap portal. Pengguna Anda masuk ke portal menggunakan AWS IAM Identity Center akun, sehingga Anda dapat menggunakan toko identitas yang ada atau toko yang dikelola oleh AWS.

Anda, dan pengguna Anda dengan izin yang memadai, dapat membuat dasbor di setiap proyek untuk memvisualisasikan data industri Anda dengan cara yang berarti. Kemudian, pengguna Anda dapat melihat dasbor ini untuk dengan cepat mendapatkan wawasan tentang data Anda dan memantau operasi Anda. Anda dapat mengonfigurasi izin administratif atau hanya baca untuk setiap proyek untuk setiap pengguna di perusahaan Anda. Untuk informasi selengkapnya, lihat [Memantau data dengan AWS IoT SiteWise Monitor](#).

Sepanjang tutorial, Anda meningkatkan AWS IoT SiteWise demo, menyediakan contoh dataset untuk ladang angin. Anda mengonfigurasi portal di SiteWise Monitor, membuat proyek, dan dasbor untuk memvisualisasikan data ladang angin. Tutorial ini juga mencakup pembuatan pengguna tambahan, bersama dengan penetapan izin untuk memiliki atau melihat proyek dan dasbor terkait.

 Note

Saat Anda menggunakan SiteWise Monitor, Anda dikenakan biaya per pengguna yang masuk ke portal (per bulan). Dalam tutorial ini, Anda membuat tiga pengguna, tetapi Anda hanya perlu masuk dengan satu pengguna. Setelah Anda menyelesaikan tutorial ini, Anda dikenakan biaya untuk satu pengguna. Untuk informasi selengkapnya, silakan lihat [Harga AWS IoT SiteWise](#).

Topik

- [Prasyarat](#)
- [Langkah 1: Buat portal di SiteWise Monitor](#)
- [Langkah 2: Masuk ke portal](#)
- [Langkah 3: Buat proyek ladang angin](#)
- [Langkah 4: Buat dasbor untuk memvisualisasikan data ladang angin](#)
- [Langkah 5: Jelajahi portal](#)
- [Langkah 6: Bersihkan sumber daya setelah tutorial](#)

Prasyarat

Untuk menyelesaikan tutorial ini, Anda memerlukan hal berikut:

- Sebuah Akun AWS. Jika Anda tidak memilikinya, lihat [Siapkan AWS akun](#).
- Komputer pengembangan yang berjalan Windows, macOS, Linux, atau Unix untuk mengakses AWS Management Console. Untuk informasi selengkapnya, lihat [Apa itu AWS Management Console?](#).
- Pengguna AWS Identity and Access Management (IAM) dengan izin administrator.
- Demo ladang AWS IoT SiteWise angin yang sedang berjalan. Saat Anda mengatur demo, itu mendefinisikan model dan aset AWS IoT SiteWise dan mengalirkan data ke mereka untuk mewakili ladang angin. Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise demo](#).
- Jika Anda mengaktifkan Pusat Identitas IAM di akun Anda, masuk ke akun AWS Organizations manajemen Anda. Untuk informasi lebih lanjut, lihat [AWS Organizations terminologi dan konsep](#). Jika Anda belum mengaktifkan IAM Identity Center, Anda akan mengaktifkannya dalam tutorial ini dan menetapkan akun Anda sebagai akun manajemen.

Jika Anda tidak dapat masuk ke akun AWS Organizations manajemen Anda, Anda dapat menyelesaikan sebagian tutorial selama Anda memiliki pengguna Pusat Identitas IAM di organisasi Anda. Dalam hal ini, Anda dapat membuat portal dan dasbor, tetapi Anda tidak dapat membuat pengguna Pusat Identitas IAM baru untuk ditetapkan ke proyek.

Langkah 1: Buat portal di SiteWise Monitor

Dalam prosedur ini, Anda membuat portal di AWS IoT SiteWise Monitor. Setiap portal adalah aplikasi web terkelola yang dapat Anda dan pengguna masuk dengan AWS IAM Identity Center akun. Dengan IAM Identity Center, Anda dapat menggunakan toko identitas perusahaan Anda yang ada atau membuatnya dikelola oleh AWS. Karyawan perusahaan Anda dapat masuk tanpa membuat terpisah Akun AWS.

Untuk membuat portal

1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.
2. Tinjau [AWS IoT SiteWise titik akhir dan kuota](#) di mana AWS IoT SiteWise didukung dan alihkan Wilayah, jika diperlukan. Anda harus menjalankan AWS IoT SiteWise demo di Wilayah yang sama.
3. Di panel navigasi kiri, pilih Portal.
4. Pilih Buat portal.
5. Jika Anda sudah mengaktifkan IAM Identity Center, lewati ke langkah 6. Jika tidak, selesaikan langkah-langkah berikut untuk mengaktifkan Pusat Identitas IAM:
 - a. Pada halaman Aktifkan AWS IAM Identity Center (SSO), masukkan alamat Email Anda, Nama depan, dan Nama belakang untuk membuat pengguna Pusat Identitas IAM bagi Anda untuk menjadi administrator portal. Gunakan alamat email yang dapat Anda akses sehingga Anda dapat menerima email untuk mengatur kata sandi bagi pengguna Pusat Identitas IAM baru Anda.

Di portal, administrator portal membuat proyek dan menugaskan pengguna ke proyek. Anda dapat membuat lebih banyak pengguna nanti.

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 **Create user**

- b. Pilih Create user (Buat pengguna).
6. Pada halaman konfigurasi Portal, selesaikan langkah-langkah berikut:
- a. Masukkan nama untuk portal Anda, seperti **WindFarmPortal**.
 - b. (Opsional) Masukkan deskripsi untuk portal Anda. Jika Anda memiliki beberapa portal, gunakan deskripsi yang bermakna untuk melacak isi setiap portal.
 - c. (Opsional) Unggah gambar untuk ditampilkan di portal.
 - d. Masukkan alamat email yang dapat dihubungi pengguna portal ketika mereka memiliki masalah dengan portal dan memerlukan bantuan dari AWS administrator perusahaan Anda untuk menyelesaikannya.
 - e. Pilih Buat portal.
7. Pada halaman Undang administrator, Anda dapat menetapkan pengguna Pusat Identitas IAM ke portal sebagai administrator. Administrator portal mengelola izin dan proyek dalam portal. Di halaman ini, lakukan hal berikut:
- a. Pilih pengguna untuk menjadi administrator portal. Jika Anda mengaktifkan IAM Identity Center sebelumnya dalam tutorial ini, pilih pengguna yang Anda buat.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Invite administrators

Select the users that you want to be portal administrators. When invited, portal administrators control users' access to the operational data of your Sitewise assets. [Learn more](#)

Send invite to selected users

Users (1)

Create user

Find resources

	Display name	Email
☒	John Doe	john.doe@example.com

Selected users (1)

Cancel Next

- b. (Optional) Pilih Kirim undangan ke pengguna yang dipilih. Klien email Anda terbuka, dan undangan muncul di badan pesan. Anda dapat menyesuaikan email sebelum mengirimkannya ke administrator portal Anda. Anda juga dapat mengirim email ke administrator portal Anda nanti. Jika Anda mencoba SiteWise Monitor untuk pertama kalinya dan akan menjadi administrator portal, Anda tidak perlu mengirim email sendiri.
 - c. Pilih Berikutnya.
8. Pada halaman Tetapkan pengguna, Anda dapat menetapkan pengguna Pusat Identitas IAM ke portal. Administrator portal nantinya dapat menetapkan pengguna ini sebagai pemilik proyek atau pemirsa. Pemilik proyek dapat membuat dasbor dalam proyek. Pemirsa proyek memiliki akses hanya-baca ke proyek yang ditetapkan. Di halaman ini, Anda dapat membuat pengguna IAM Identity Center untuk ditambahkan ke portal.

Note

Jika Anda tidak masuk ke akun AWS Organizations manajemen, Anda tidak dapat membuat pengguna Pusat Identitas IAM. Pilih Tetapkan pengguna untuk membuat portal tanpa pengguna portal, lalu lewati langkah ini.

Di halaman ini, lakukan hal berikut:

- a. Selesaikan langkah-langkah berikut dua kali untuk membuat dua pengguna IAM Identity Center:

- i. Pilih Buat pengguna untuk membuka kotak dialog tempat Anda memasukkan detail untuk pengguna baru.
- ii. Masukkan alamat Email, Nama depan, dan Nama belakang untuk pengguna baru. IAM Identity Center mengirimkan email kepada pengguna untuk mengatur kata sandi mereka. Jika Anda ingin masuk ke portal sebagai pengguna ini, pilih alamat email yang dapat Anda akses. Setiap alamat email harus unik. Pengguna Anda masuk ke portal menggunakan alamat email mereka sebagai nama pengguna mereka.

Create user [X]

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

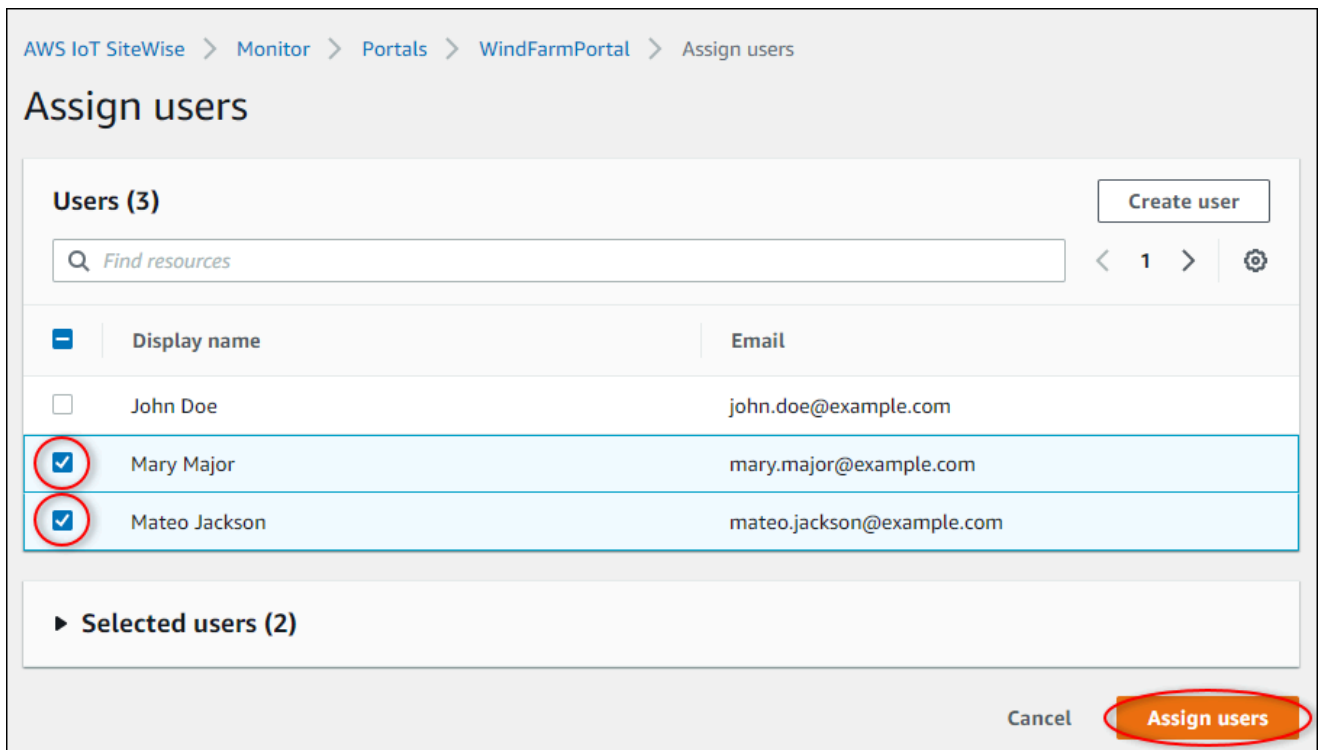
Email address
mary.major@example.com

First name
Mary

Last name
Major

Cancel Create user

- iii. Pilih Create user (Buat pengguna).
- b. Pilih dua pengguna IAM Identity Center yang Anda buat pada langkah sebelumnya.



- c. Pilih Tetapkan pengguna untuk menambahkan pengguna ini ke portal.

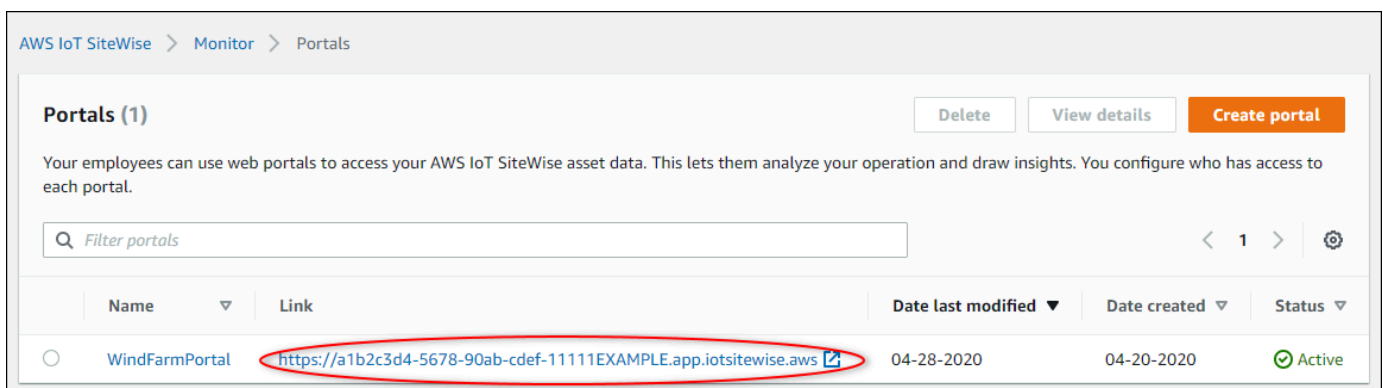
Halaman portal terbuka dengan portal baru Anda terdaftar.

Langkah 2: Masuk ke portal

Dalam prosedur ini, Anda masuk ke portal baru Anda menggunakan AWS IAM Identity Center pengguna yang Anda tambahkan ke portal.

Untuk masuk ke portal

1. Pada halaman Portal, pilih Tautan portal baru Anda untuk membuka portal Anda di tab baru.



2. Jika Anda membuat pengguna IAM Identity Center pertama Anda di awal tutorial, gunakan langkah-langkah berikut untuk membuat kata sandi untuk pengguna Anda:
 - a. Periksa email Anda untuk baris subjek Invitation to join AWS IAM Identity Center.
 - b. Buka email undangan itu dan pilih Accept invitation.
 - c. Di jendela baru, atur kata sandi untuk pengguna Pusat Identitas IAM Anda.

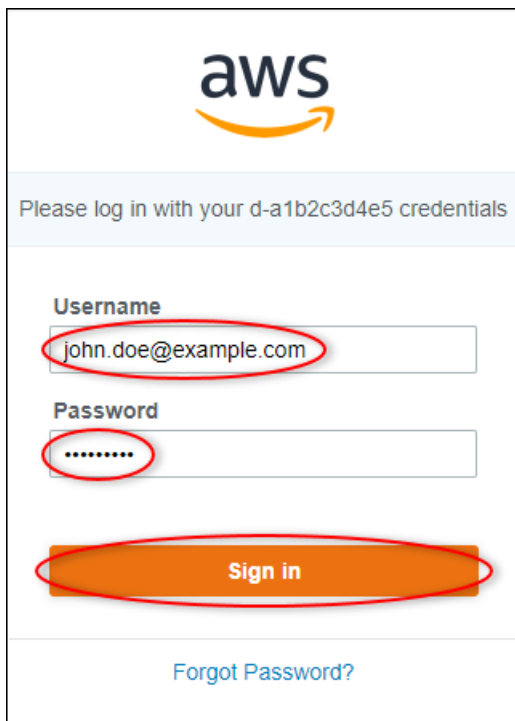
Jika Anda ingin masuk nanti ke portal sebagai pengguna Pusat Identitas IAM kedua dan ketiga yang Anda buat sebelumnya, Anda juga dapat menyelesaikan langkah-langkah ini untuk mengatur kata sandi bagi pengguna tersebut.

 Note

Jika Anda tidak menerima email, Anda dapat membuat kata sandi untuk pengguna Anda di konsol Pusat Identitas IAM. Untuk informasi selengkapnya, lihat [Mengatur ulang kata sandi pengguna IAM Identity Center untuk pengguna akhir di Panduan AWS IAM Identity Center](#) Pengguna.

3. Masukkan Pusat Identitas IAM Anda Username dan Password. Jika Anda membuat pengguna IAM Identity Center Anda sebelumnya dalam tutorial ini, Username adalah alamat email pengguna administrator portal yang Anda buat.

Semua pengguna portal, termasuk administrator portal, harus masuk dengan kredensial pengguna Pusat Identitas IAM mereka. Kredensial ini biasanya bukan kredensial yang sama yang Anda gunakan untuk masuk ke AWS Management Console



aws

Please log in with your d-a1b2c3d4e5 credentials

Username
john.doe@example.com

Password
.....

Sign in

[Forgot Password?](#)

4. Pilih Sign in.

Portal Anda terbuka.

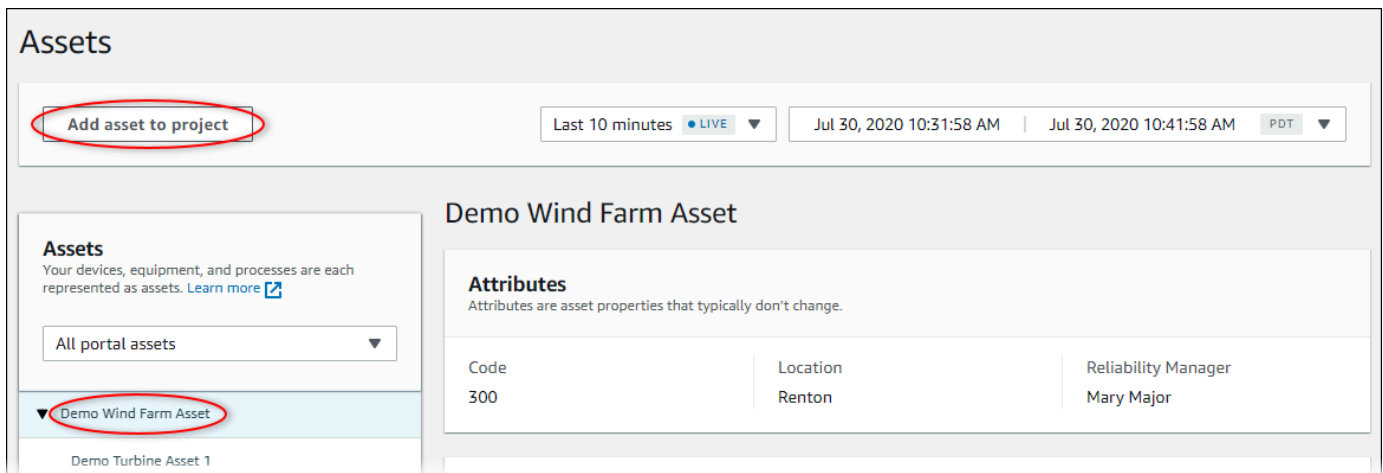
Langkah 3: Buat proyek ladang angin

Dalam prosedur ini, Anda membuat proyek di portal Anda. Proyek adalah sumber daya yang menentukan sekumpulan izin, aset, dan dasbor, yang dapat Anda konfigurasi untuk memvisualisasikan data aset dalam proyek tersebut. Dengan proyek, Anda menentukan siapa yang memiliki akses ke subset operasi Anda dan bagaimana data subset tersebut divisualisasikan. Anda dapat menetapkan pengguna portal sebagai pemilik atau pemirsa setiap proyek. Pemilik proyek dapat membuat dasbor untuk memvisualisasikan data dan berbagi proyek dengan pengguna lain. Pemirsa proyek dapat melihat dasbor tetapi tidak mengeditnya. Untuk informasi selengkapnya tentang peran di SiteWise Monitor, lihat [SiteWise Memantau peran](#).

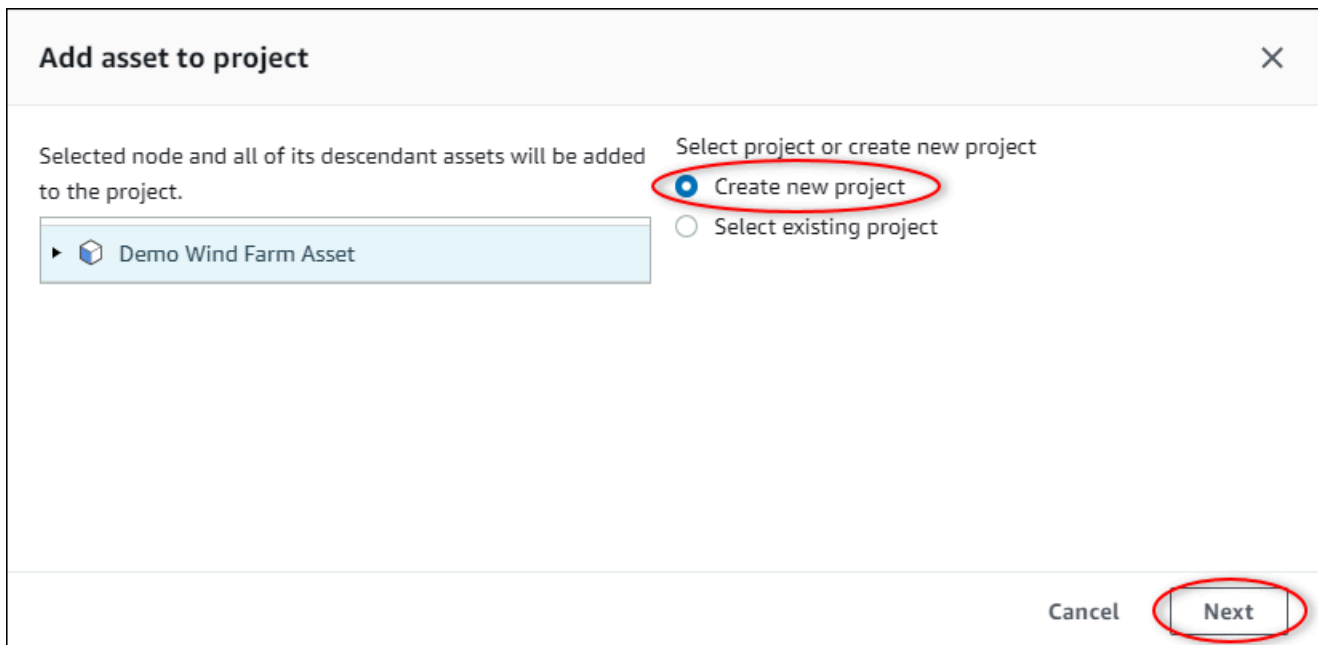
Untuk membuat proyek ladang angin

1. Di panel navigasi kiri di portal Anda, pilih tab Aset. Di halaman Aset, Anda dapat menjelajahi semua aset yang tersedia di portal dan menambahkan aset ke proyek.

2. Di browser aset, pilih Demo Wind Farm Asset. Saat memilih aset, Anda dapat menjelajahi data langsung dan historis aset tersebut. Anda juga dapat menekan Shift untuk memilih beberapa aset dan membandingkan datanya side-by-side.
3. Pilih Tambahkan aset ke proyek di kiri atas. Proyek berisi dasbor yang dapat dilihat pengguna portal Anda untuk menjelajahi data Anda. Setiap proyek memiliki akses ke subset aset Anda di AWS IoT SiteWise. Saat Anda menambahkan aset ke proyek, semua pengguna yang memiliki akses ke proyek tersebut juga dapat mengakses data untuk aset tersebut dan anak-anaknya.



4. Dalam kotak dialog Tambahkan aset ke proyek, pilih Buat proyek baru, lalu pilih Berikutnya.



5. Di kotak dialog Buat proyek baru, masukkan nama Proyek dan deskripsi Proyek untuk proyek Anda, lalu pilih Tambahkan aset ke proyek.



Create new project ✕

Project name

The project name can have up to 256 characters.

Project description

The project description can have up to 2048 characters.

Cancel Previous Add asset to project

Halaman proyek baru Anda terbuka.

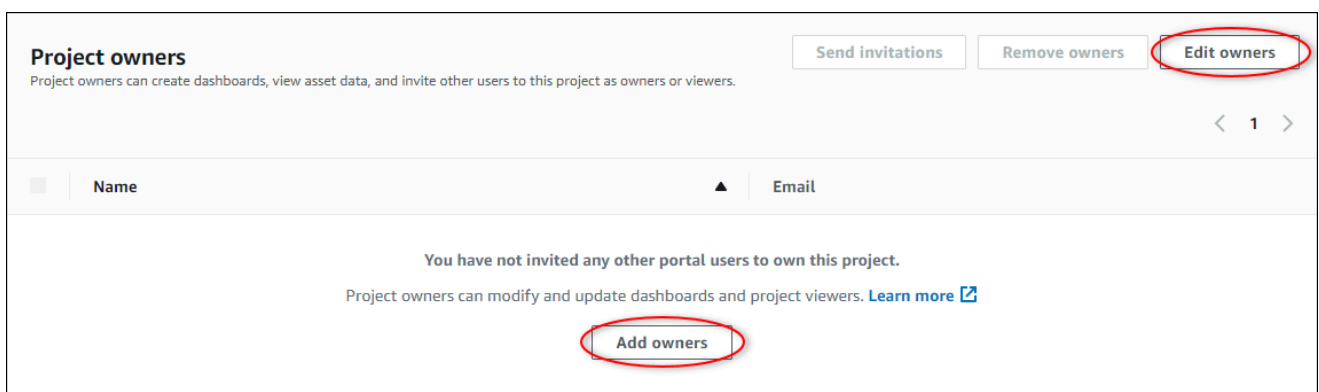
6. Pada halaman proyek, Anda dapat menambahkan pengguna portal sebagai pemilik atau pemirsa proyek ini.

Note

Jika Anda tidak masuk ke akun AWS Organizations manajemen, Anda mungkin tidak memiliki pengguna portal untuk menetapkan proyek ini, sehingga Anda dapat melewati langkah ini.

Di halaman ini, lakukan hal berikut:

- a. Di bawah Pemilik proyek, pilih Tambah pemilik atau Edit pengguna.



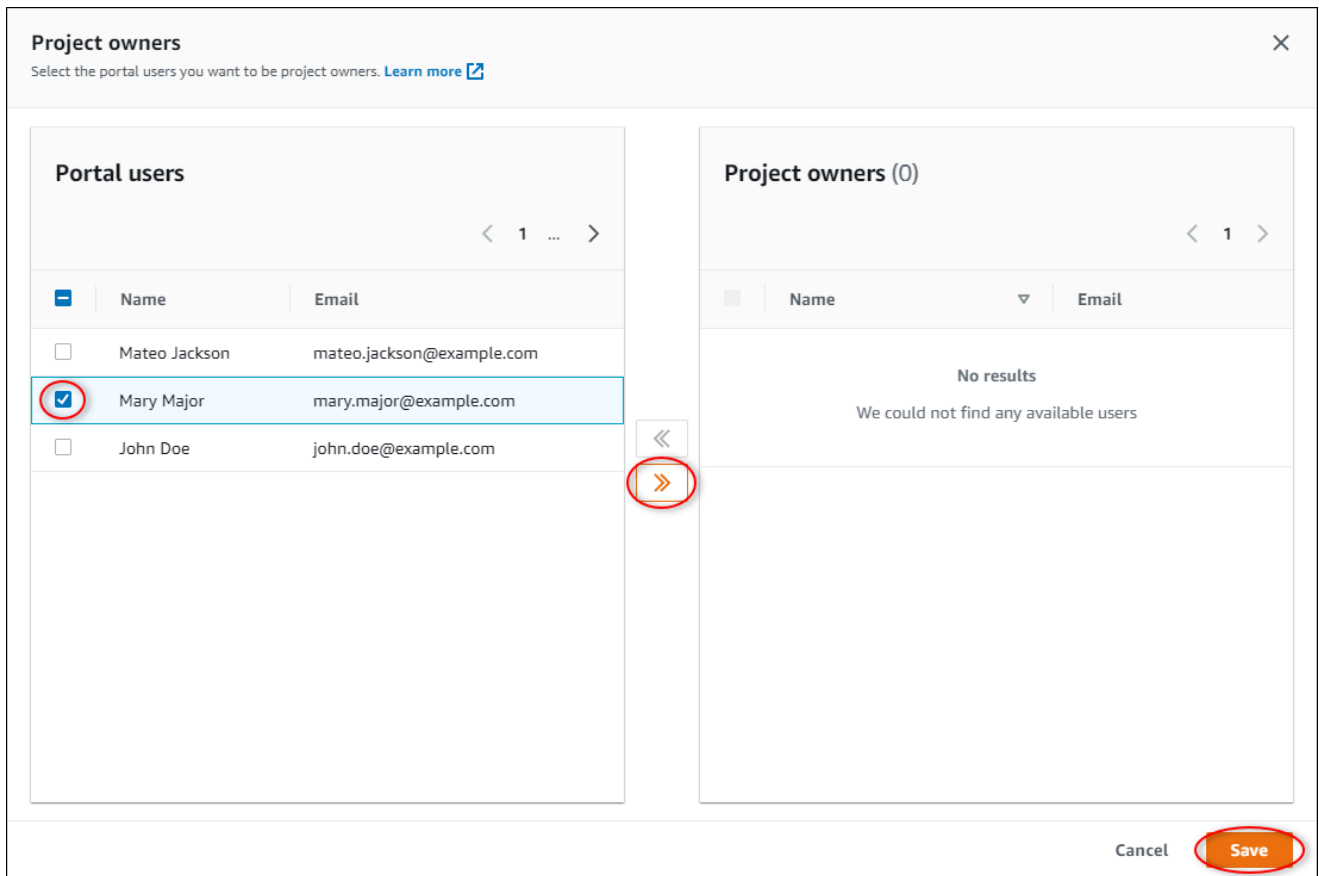
Project owners Send invitations Remove owners Edit owners

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		

- b. Pilih pengguna untuk ditambahkan sebagai pemilik proyek (misalnya, Mary Major), dan kemudian pilih ikon >>.



- c. Pilih Simpan.

Pengguna Pusat Identitas IAM Anda Mary Majordapat masuk ke portal ini untuk mengedit dasbor dalam proyek ini dan berbagi proyek ini dengan pengguna lain di portal ini.

- d. Di bawah Penampil proyek, pilih Tambahkan pemirsa atau Edit pengguna.
- e. Pilih pengguna untuk ditambahkan sebagai penampil proyek (misalnya, Mateo Jackson), dan kemudian pilih ikon >>.
- f. Pilih Simpan.

Pengguna Pusat Identitas IAM Anda Mateo Jackson dapat masuk ke portal ini untuk melihat, tetapi tidak mengedit, dasbor di proyek ladang angin.

Langkah 4: Buat dasbor untuk memvisualisasikan data ladang angin

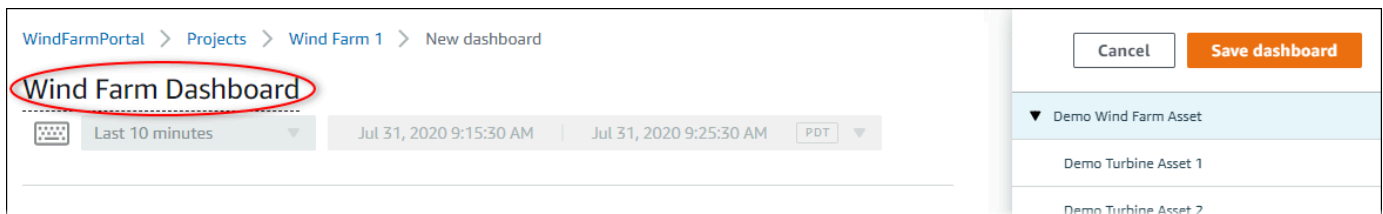
Dalam prosedur ini, Anda membuat dasbor untuk memvisualisasikan data ladang angin demo. Dasbor berisi visualisasi yang dapat disesuaikan dari data aset proyek Anda. Setiap visualisasi dapat memiliki jenis yang berbeda, seperti diagram garis, diagram batang, atau tampilan indikator kinerja kunci (KPI). Anda dapat memilih jenis visualisasi yang paling sesuai untuk data Anda. Pemilik proyek dapat mengedit dasbor, sedangkan pemirsa proyek hanya dapat melihat dasbor untuk mendapatkan wawasan.

Untuk membuat dasbor dengan visualisasi

1. Pada halaman proyek baru Anda, pilih Buat dasbor untuk membuat dasbor dan buka halaman editnya.

Di halaman edit dasbor, Anda dapat menyeret properti aset dari hierarki aset ke dasbor untuk membuat visualisasi. Kemudian, Anda dapat mengedit setiap judul visualisasi, judul legenda, jenis, ukuran, dan lokasi di dasbor.

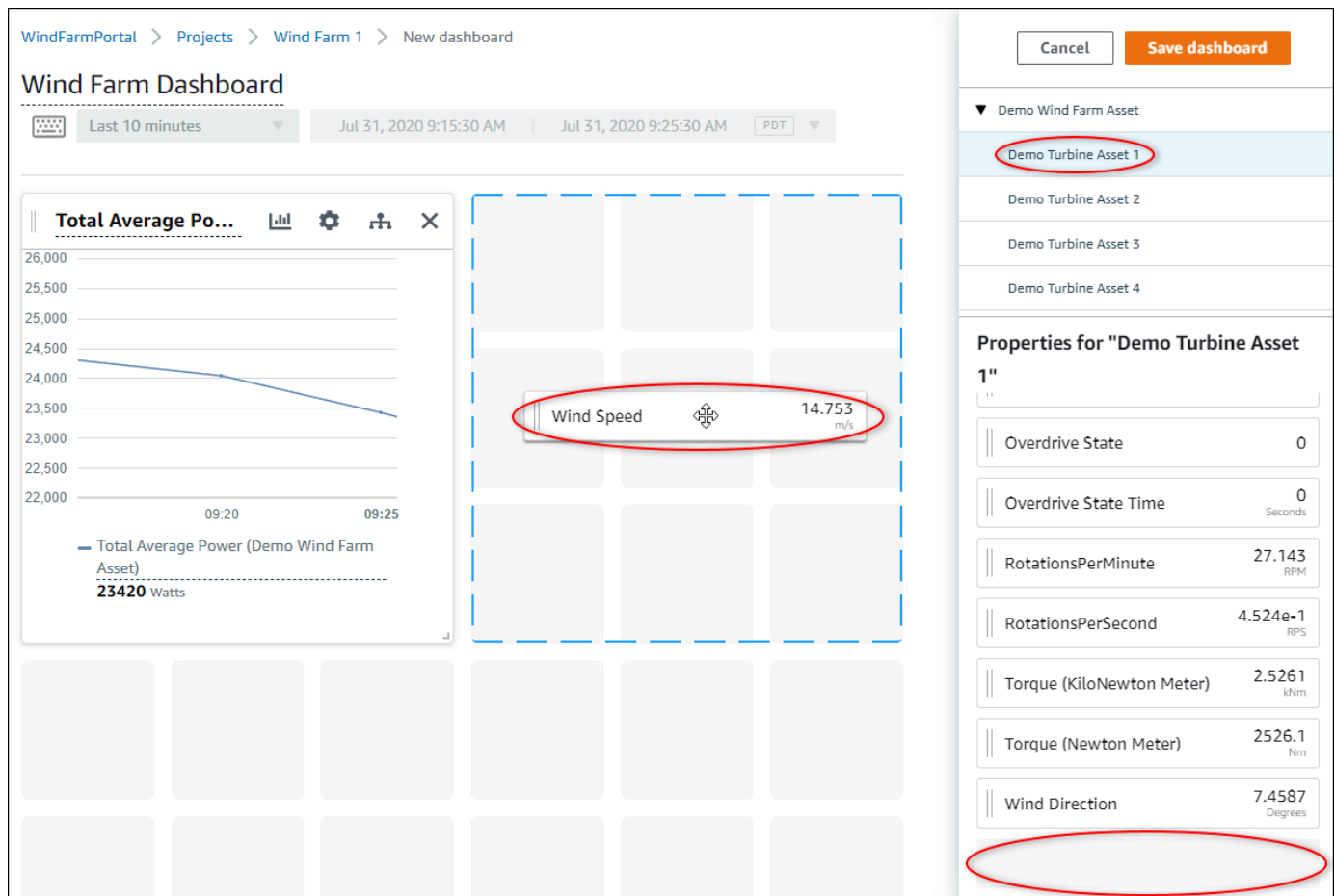
2. Masukkan nama dasbor Anda.



3. Seret Total Average Power dari Demo Wind Farm Asset ke dasbor untuk membuat visualisasi.

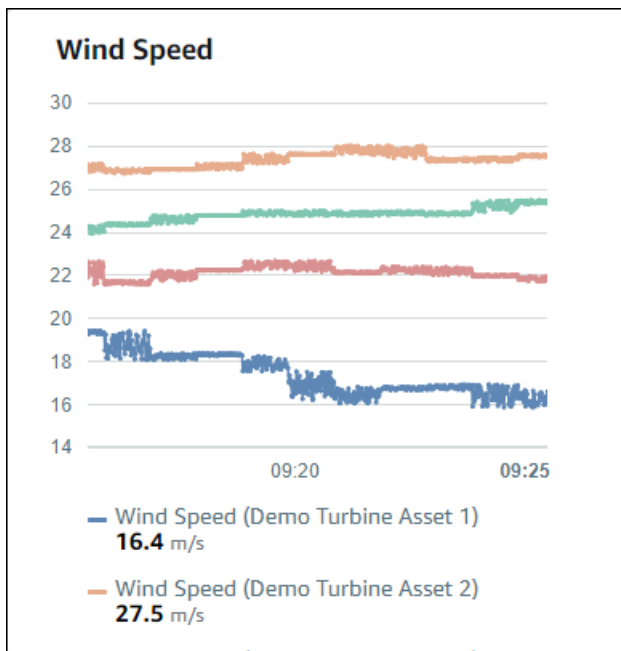
The screenshot shows the 'Wind Farm Dashboard' in the AWS IoT SiteWise console. The breadcrumb trail is 'WindFarmPortal > Projects > Wind Farm 1 > New dashboard'. The dashboard title is 'Wind Farm Dashboard'. Below the title, there are filters for 'Last 10 minutes', a date range from 'Jul 31, 2020 9:15:30 AM' to 'Jul 31, 2020 9:25:30 AM', and a time zone dropdown set to 'PDT'. The dashboard grid contains several placeholder widgets. One widget, 'Total Average Power', is highlighted with a red oval and shows a value of '24038 Watts'. The right sidebar has a 'Cancel' button and a 'Save dashboard' button. Below these is a section titled 'Demo Wind Farm Asset' with a list of assets: 'Demo Turbine Asset 1', 'Demo Turbine Asset 2', 'Demo Turbine Asset 3', and 'Demo Turbine Asset 4'. Below the asset list is a section titled 'Properties for "Demo Wind Farm Asset"' with two properties: 'Code' with a value of '300' and 'Total Overdrive State Time' with a value of '0 seconds'. The 'Total Overdrive State Time' property is highlighted with a red oval.

4. Pilih Demo Turbine Asset 1 untuk menampilkan properti untuk aset itu, lalu seret Wind Speed ke dasbor untuk membuat visualisasi untuk kecepatan angin.

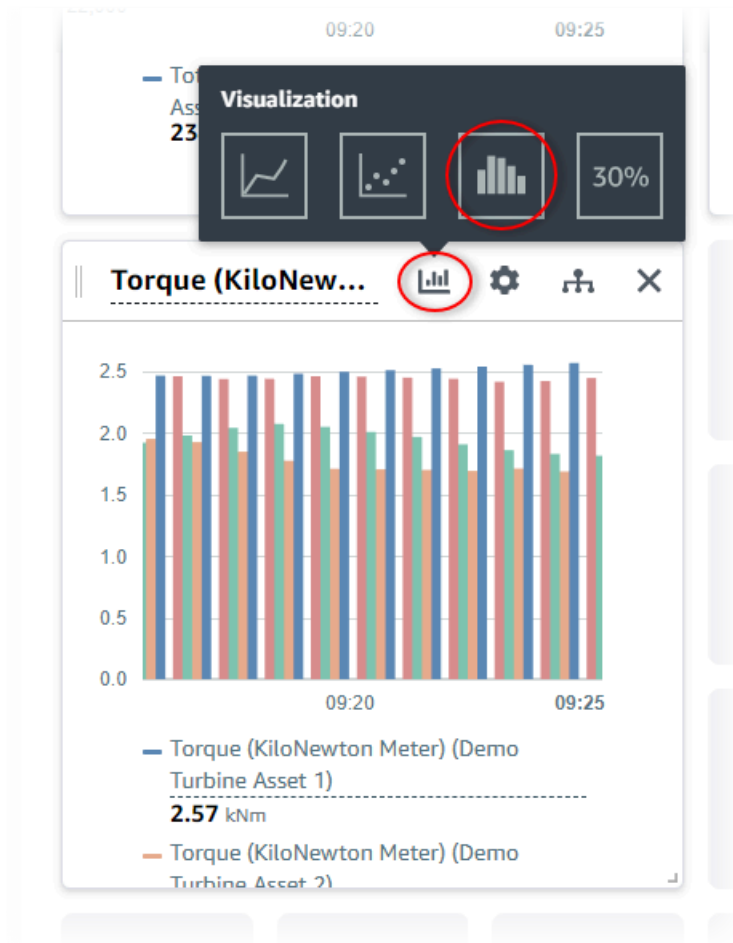


5. Tambahkan Wind Speed ke visualisasi kecepatan angin baru untuk masing-masing Demo Turbine Asset 2, 3, dan 4 (dalam urutan itu).

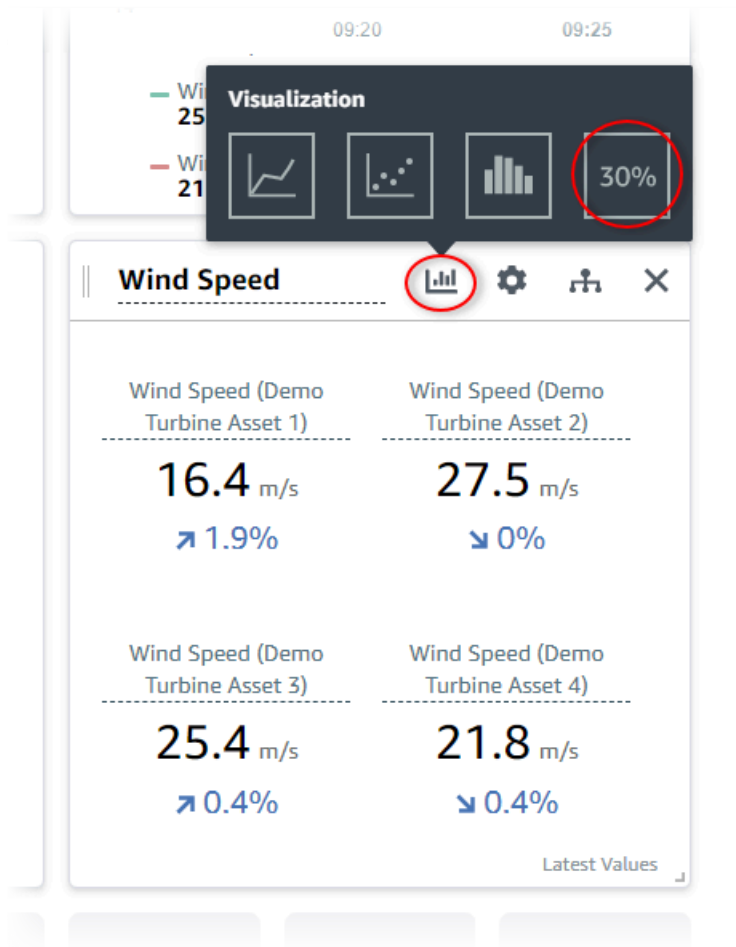
Anda Wind Speed visualisasi akan terlihat mirip dengan tangkapan layar berikut.



6. Ulangi langkah 4 dan 5 untuk turbin angin Torque (KiloNewton Meter) properti untuk membuat visualisasi untuk torsi turbin angin.
7. Pilih ikon jenis visualisasi untuk Torque (KiloNewton Meter) visualisasi, lalu pilih ikon diagram batang.

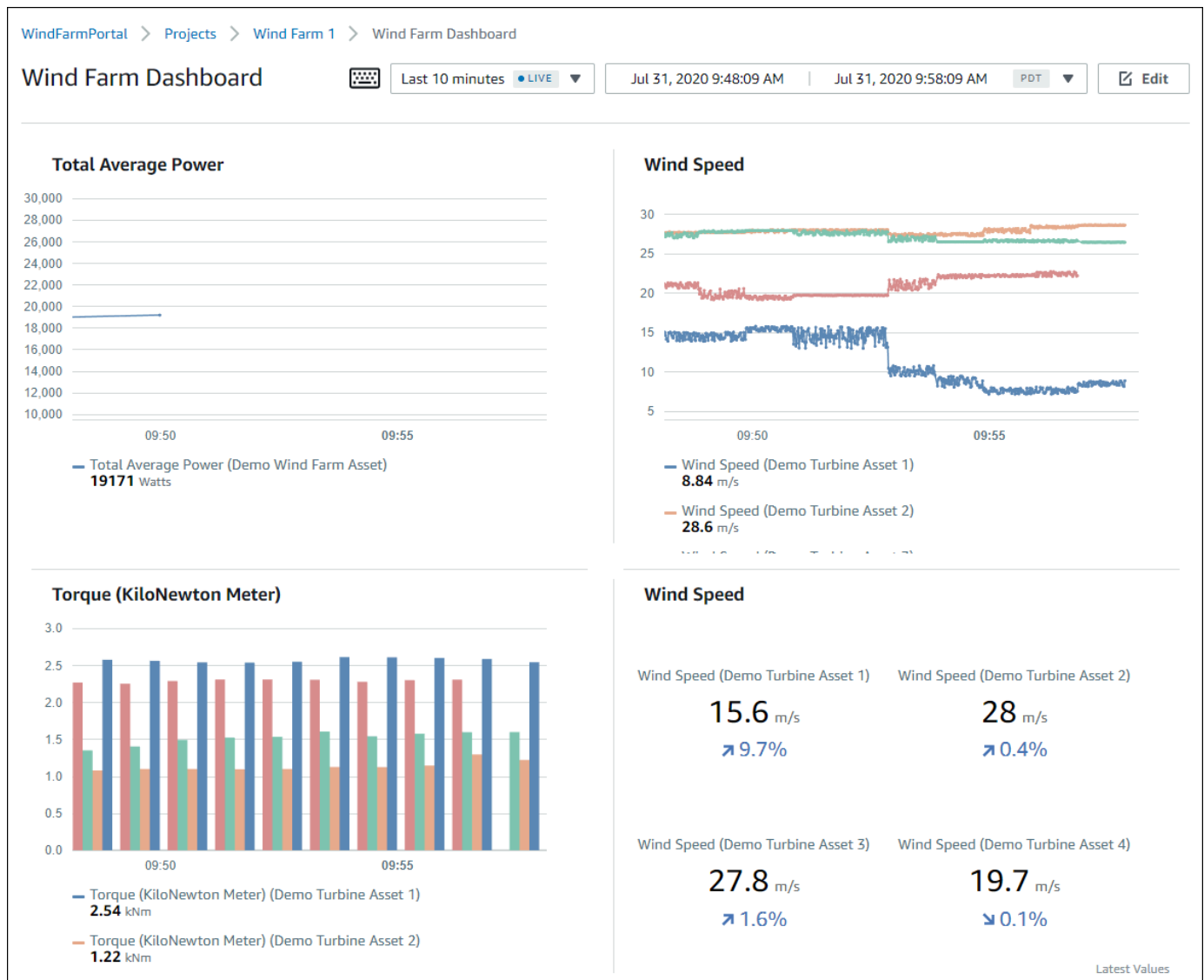


8. Ulangi langkah 4 dan 5 untuk turbin angin Wind Direction properti untuk membuat visualisasi untuk arah angin.
9. Pilih ikon jenis visualisasi untuk Wind Direction visualisasi, lalu pilih ikon bagan KPI (30%).



10. (Opsional) Buat perubahan lain pada setiap judul visualisasi, judul legenda, jenis, ukuran, dan lokasi sesuai kebutuhan.
11. Pilih Simpan dasbor di kanan atas untuk menyimpan dasbor Anda.

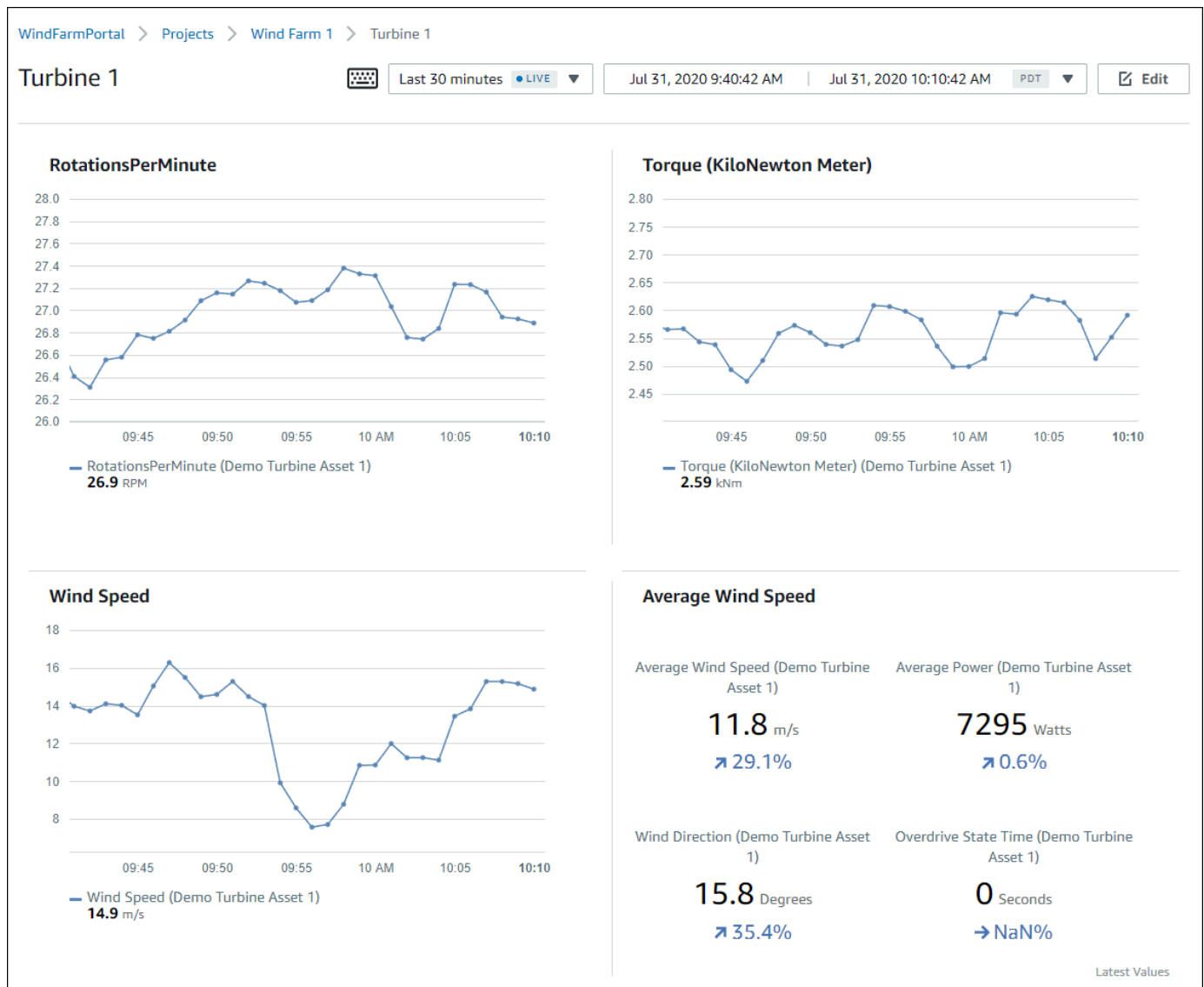
Dasbor Anda akan terlihat mirip dengan tangkapan layar berikut.



12. (Optional) Buat dasbor tambahan untuk setiap aset turbin angin.

Sebagai praktik terbaik, kami menyarankan Anda membuat dasbor untuk setiap aset sehingga pemirsa proyek Anda dapat menyelidiki masalah apa pun dengan setiap aset individu. Anda hanya dapat menambahkan hingga 5 aset ke setiap visualisasi, jadi Anda harus membuat beberapa dasbor untuk aset hierarkis Anda dalam banyak skenario.

Dasbor untuk turbin angin demo mungkin terlihat mirip dengan tangkapan layar berikut.



13. (Opsional) Ubah timeline atau pilih titik data pada visualisasi untuk menjelajahi data di dasbor Anda. Untuk informasi selengkapnya, lihat [Melihat dasbor](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.

Langkah 5: Jelajahi portal

Dalam prosedur ini, Anda dapat menjelajahi portal sebagai pengguna dengan izin lebih sedikit daripada administrator AWS IoT SiteWise portal.

Untuk menjelajahi portal dan menyelesaikan tutorial

- (Opsional) Jika Anda menambahkan pengguna lain ke proyek sebagai pemilik atau pemirsa, Anda dapat masuk ke portal sebagai pengguna ini. Ini memungkinkan Anda menjelajahi portal sebagai pengguna dengan izin lebih sedikit daripada administrator portal.

Important

Anda dikenakan biaya untuk setiap pengguna yang masuk ke portal. Untuk informasi selengkapnya, silakan lihat [Harga AWS IoT SiteWise](#).

Untuk menjelajahi portal sebagai pengguna lain, lakukan hal berikut:

- a. Pilih Keluar di kiri bawah portal untuk keluar dari aplikasi web.
- b. Pilih Keluar di kanan atas portal aplikasi Pusat Identitas IAM untuk keluar dari pengguna Pusat Identitas IAM Anda.
- c. Masuk ke portal sebagai pengguna Pusat Identitas IAM yang Anda tetapkan sebagai pemilik proyek atau penampil proyek. Untuk informasi selengkapnya, lihat [Langkah 2: Masuk ke portal](#).

Anda telah menyelesaikan tutorial. Ketika Anda selesai menjelajahi ladang angin demo Anda di SiteWise Monitor, ikuti prosedur selanjutnya untuk membersihkan sumber daya Anda.

Langkah 6: Bersihkan sumber daya setelah tutorial

Setelah Anda menyelesaikan tutorial, Anda dapat membersihkan sumber daya Anda. Anda tidak dikenakan biaya AWS IoT SiteWise jika pengguna tidak masuk ke portal Anda, tetapi Anda dapat menghapus portal dan Direktori Pusat Identitas AWS IAM pengguna Anda. Aset ladang angin demo Anda dihapus pada akhir durasi yang Anda pilih saat Anda membuat demo, atau Anda dapat menghapus demo secara manual. Untuk informasi selengkapnya, lihat [Hapus AWS IoT SiteWise demo](#).

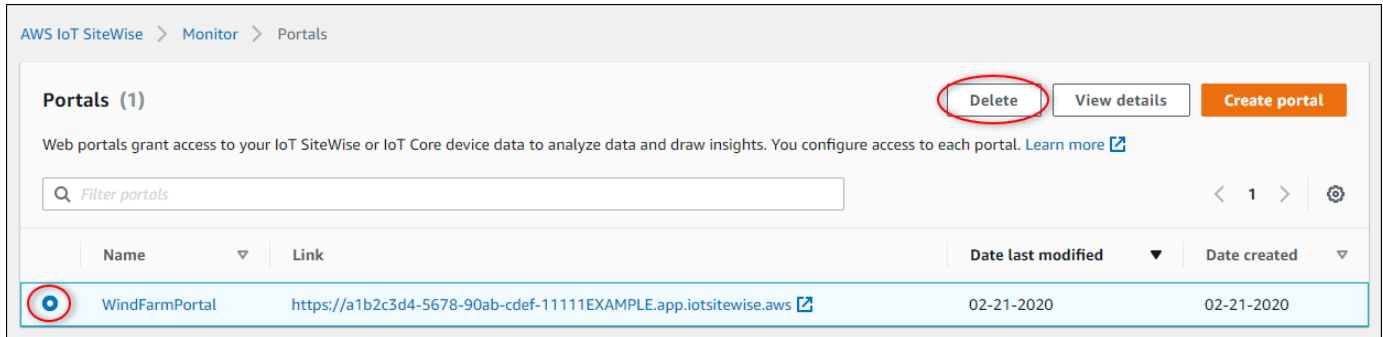
Gunakan prosedur berikut untuk menghapus portal Anda dan pengguna Pusat Identitas IAM.

Untuk menghapus portal

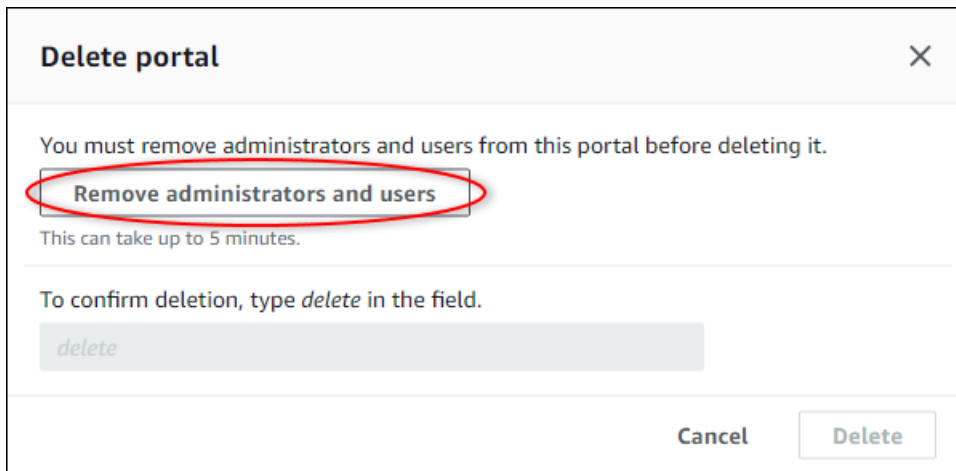
1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di panel navigasi kiri, pilih Portal.
3. Pilih portal Anda WindFarmPortal, lalu pilih Hapus.

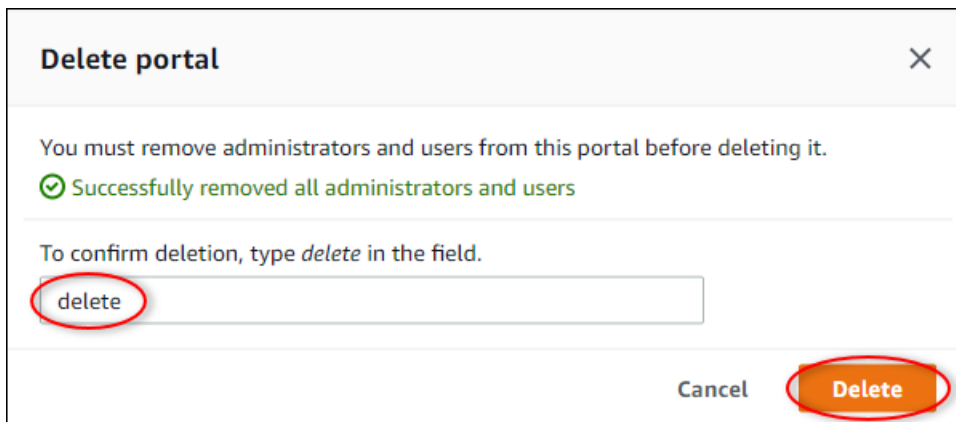
Saat Anda menghapus portal atau proyek, aset yang terkait dengan proyek yang dihapus tidak akan terpengaruh.



4. Di kotak dialog Hapus portal, pilih Hapus administrator dan pengguna.

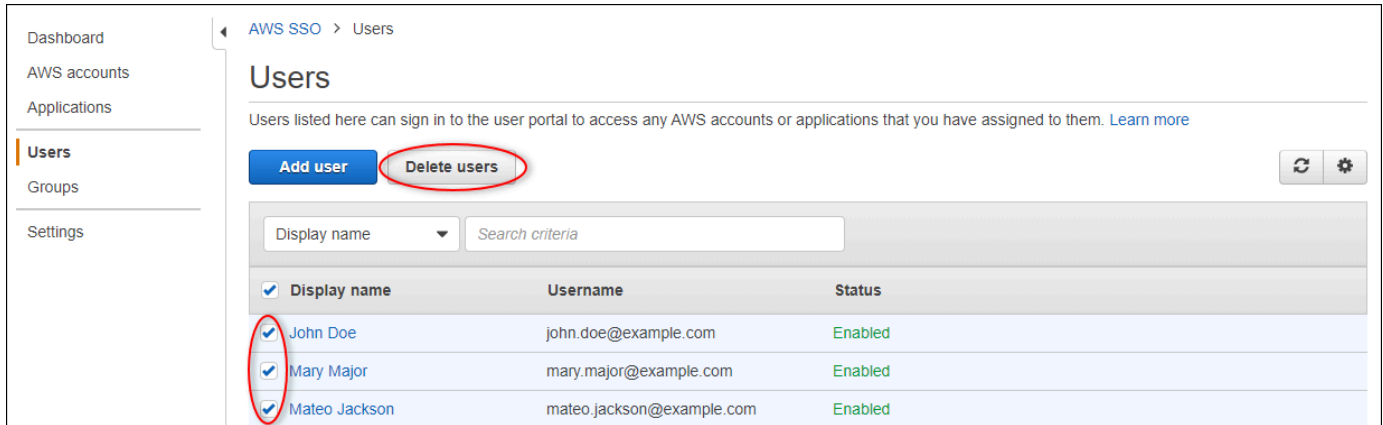


5. Masukkan **delete** untuk mengonfirmasi penghapusan, lalu pilih Hapus.

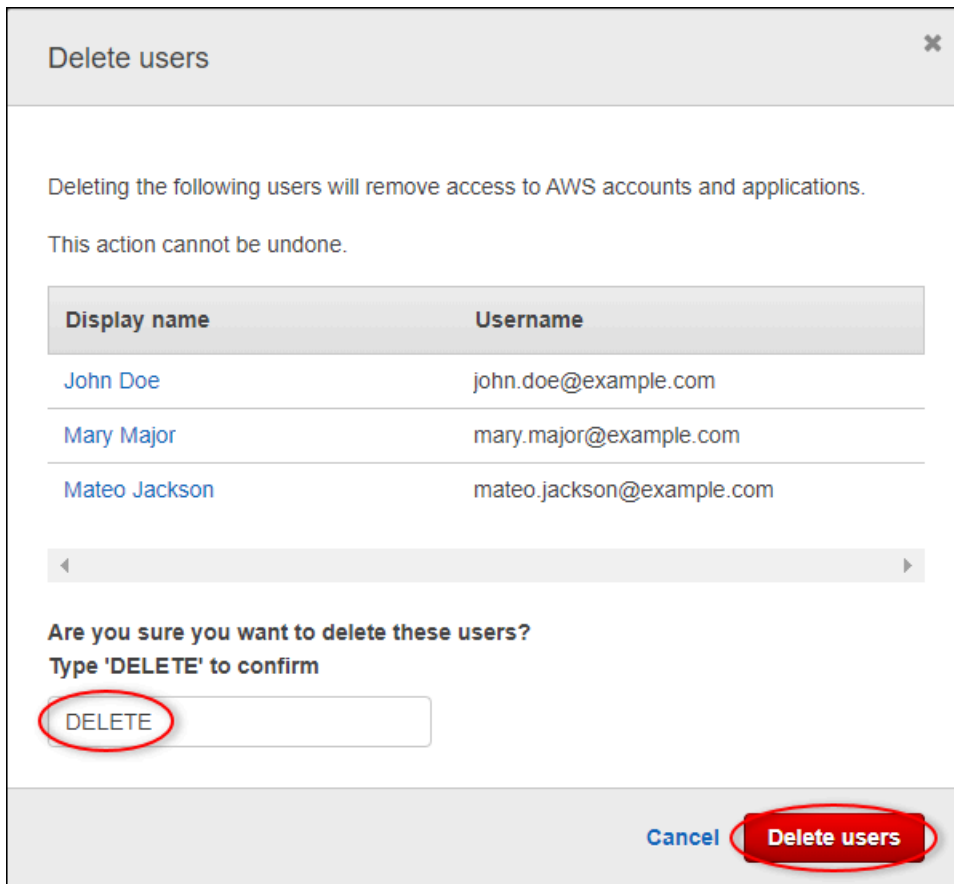


Untuk menghapus pengguna IAM Identity Center

1. Arahkan ke [konsol Pusat Identitas IAM](#).
2. Pada panel navigasi kiri, pilih Pengguna.
3. Pilih kotak centang untuk menghapus setiap pengguna, lalu pilih Hapus pengguna.



4. Di kotak dialog Hapus pengguna, masukkan **DELETE**, lalu pilih Hapus pengguna.



Publikasikan pembaruan nilai properti ke Amazon DynamoDB

Tutorial ini memperkenalkan cara mudah untuk menyimpan data Anda dengan menggunakan [Amazon DynamoDB](#), sehingga lebih mudah untuk mengakses data aset historis tanpa berulang kali menanyakan API. AWS IoT SiteWise Setelah Anda menyelesaikan tutorial ini, Anda dapat membuat perangkat lunak khusus yang mengkonsumsi data aset Anda, seperti peta langsung kecepatan angin dan arah di seluruh ladang angin. Jika Anda ingin memantau dan memvisualisasikan data Anda tanpa menerapkan solusi perangkat lunak khusus, lihat [Memantau data dengan AWS IoT SiteWise Monitor](#).

Dalam tutorial ini, Anda membangun AWS IoT SiteWise demo yang menyediakan kumpulan sampel data untuk ladang angin. Anda mengonfigurasi pembaruan nilai properti dari demo ladang angin untuk mengirim data, melalui aturan AWS IoT Core, ke tabel DynamoDB yang Anda buat. Saat Anda mengaktifkan pembaruan nilai properti, AWS IoT SiteWise mengirimkan data Anda ke AWS IoT Core dalam pesan MQTT. Kemudian, tentukan aturan AWS IoT Core yang melakukan tindakan, seperti tindakan DynamoDB, tergantung pada isi pesan tersebut. Untuk informasi selengkapnya, lihat [Berinteraksi dengan AWS layanan lain](#).

Topik

- [Prasyarat](#)
- [Langkah 1: Konfigurasi AWS IoT SiteWise untuk mempublikasikan pembaruan nilai properti](#)
- [Langkah 2: Buat aturan di AWS IoT Core](#)
- [Langkah 3: Buat tabel DynamoDB](#)
- [Langkah 4: Konfigurasi tindakan aturan DynamoDB](#)
- [Langkah 5: Jelajahi data di DynamoDB](#)
- [Langkah 6: Bersihkan sumber daya setelah tutorial](#)

Prasyarat

Untuk menyelesaikan tutorial ini, Anda memerlukan hal berikut:

- Sebuah AWS akun. Jika Anda tidak memilikinya, lihat [Siapkan AWS akun](#).
- Komputer pengembangan yang menjalankan Windows, macOS, Linux, atau Unix untuk mengakses file. AWS Management Console Untuk informasi lebih lanjut, lihat [Apa itu AWS Management Console?](#)

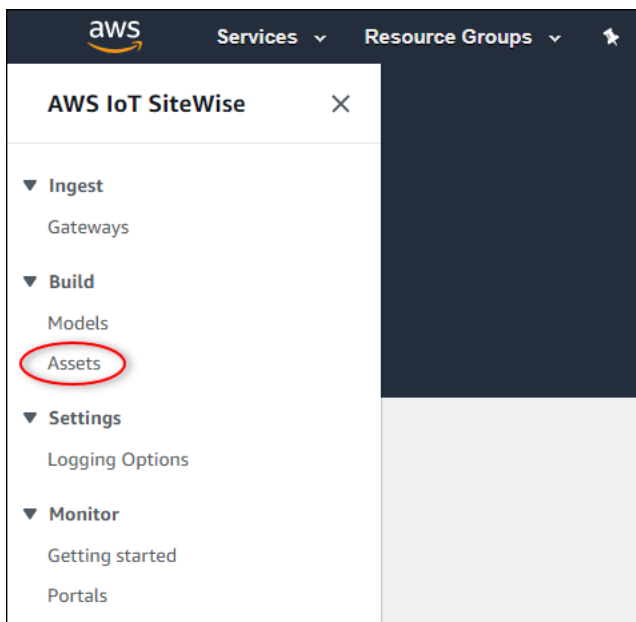
- Pengguna IAM dengan izin administrator.
- Demo peternakan AWS IoT SiteWise angin yang sedang berjalan. Saat Anda mengatur demo, itu mendefinisikan model dan aset AWS IoT SiteWise dan mengalirkan data ke mereka untuk mewakili ladang angin. Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise demo](#).

Langkah 1: Konfigurasikan AWS IoT SiteWise untuk mempublikasikan pembaruan nilai properti

Dalam prosedur ini, Anda mengaktifkan pemberitahuan nilai properti pada aset turbin demo Anda Wind Speed properti. Setelah Anda mengaktifkan pemberitahuan nilai properti, AWS IoT SiteWise menerbitkan setiap pembaruan nilai dalam pesan MQTT ke Core. AWS IoT

Untuk mengaktifkan pemberitahuan pembaruan nilai properti pada properti aset

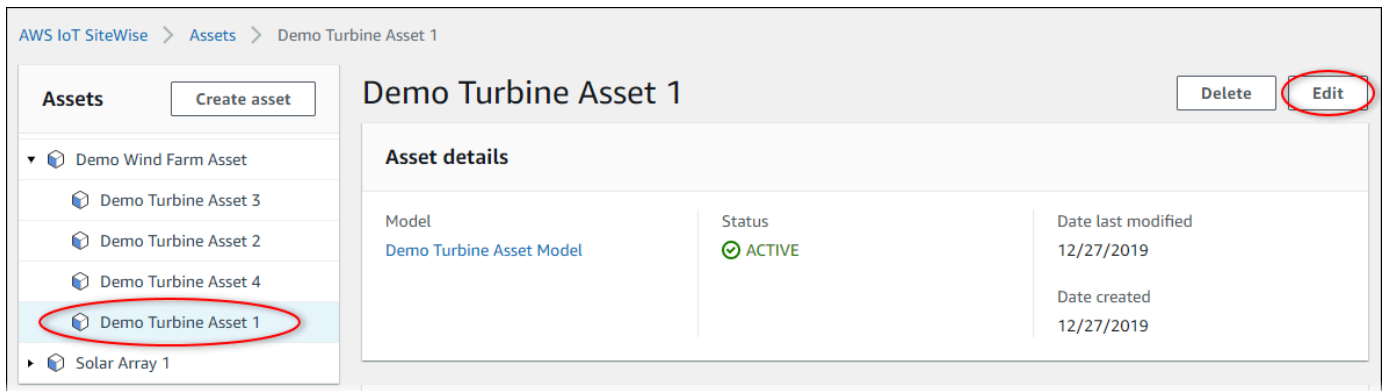
1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.
2. Tinjau [AWS IoT SiteWise titik akhir dan kuota](#) di mana AWS IoT SiteWise didukung dan alihkan AWS Wilayah, jika perlu. Beralih ke Wilayah tempat Anda menjalankan AWS IoT SiteWise demo.
3. Di panel navigasi kiri, pilih Aset.



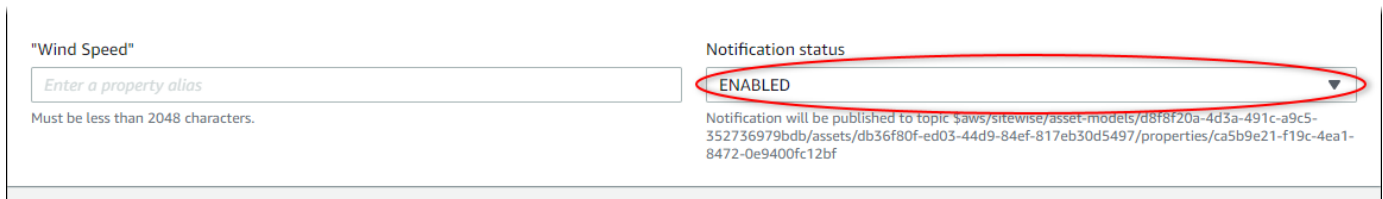
4. Pilih panah di sebelah Demo Wind Farm Asset untuk memperluas hierarki aset ladang angin.



5. Pilih turbin demo dan pilih Edit.



6. Perbarui Wind Speedstatus Pemberitahuan properti ke ENABLED.



7. Pilih Simpan aset di bagian bawah halaman.
8. Ulangi langkah 5 hingga 7 untuk setiap aset turbin demo.
9. Pilih turbin demo (misalnya, Demo Turbine Asset 1).
10. Pilih Pengukuran.
11. Pilih ikon salin di sebelah Wind Speedproperti untuk menyalin topik notifikasi ke clipboard Anda. Simpan topik notifikasi untuk digunakan nanti dalam tutorial ini. Anda hanya perlu merekam topik notifikasi dari satu turbin.

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

Topik notifikasi akan terlihat seperti contoh berikut.

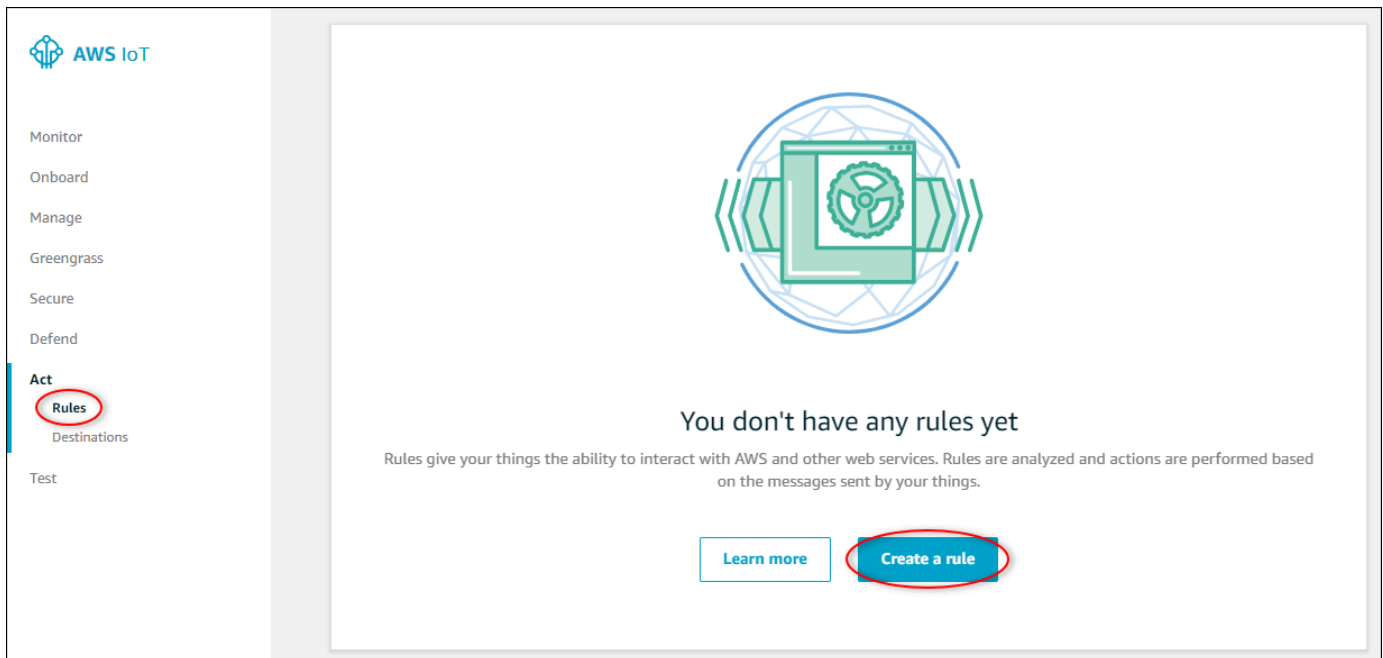
```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

Langkah 2: Buat aturan di AWS IoT Core

Dalam prosedur ini, Anda membuat aturan di AWS IoT Core yang mem-parsing pesan notifikasi nilai properti dan menyisipkan data ke dalam tabel Amazon DynamoDB. AWS IoT Aturan inti mengurai pesan MQTT dan melakukan tindakan berdasarkan konten dan topik setiap pesan. Kemudian, Anda membuat aturan dengan tindakan DynamoDB untuk menyisipkan data ke tabel DynamoDB yang Anda buat sebagai bagian dari tutorial ini.

Untuk membuat aturan dengan tindakan DynamoDB

1. Navigasikan ke [konsol AWS IoT](#) tersebut. Jika tombol Mulai muncul, pilihlah.
2. Di panel navigasi kiri, pilih Bertindak lalu pilih Aturan.



3. Jika kotak dialog Anda belum memiliki aturan apa pun muncul, pilih Buat aturan. Jika tidak, pilih Buat.
4. Masukkan nama dan deskripsi aturan.

Create a rule

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).

Name

WindSpeedRule

Description

A DynamoDBv2 rule that records wind data from wind turbine assets in AWS IoT SiteWise.

5. Temukan topik notifikasi yang Anda simpan sebelumnya dalam tutorial ini.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-1111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-3333EXAMPLE
```

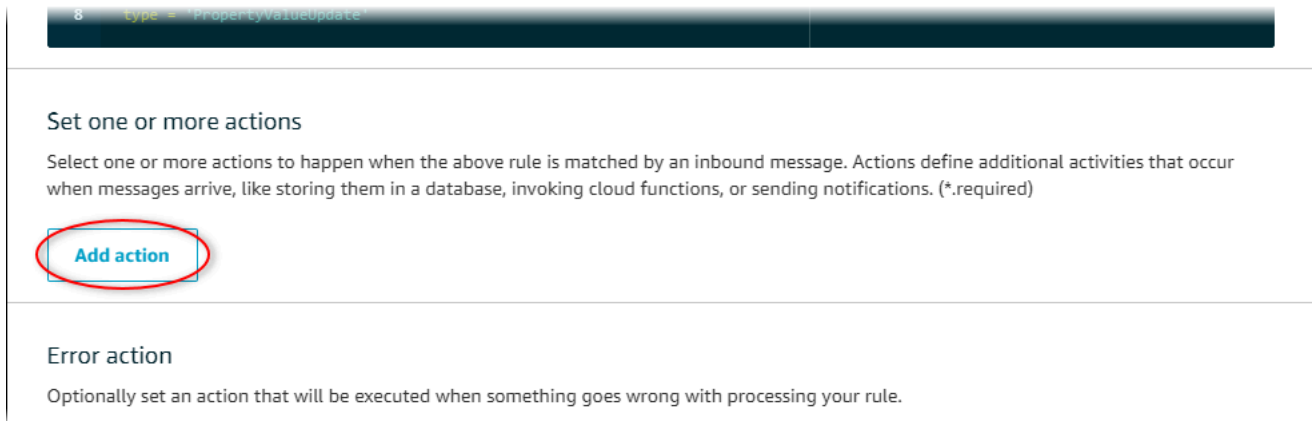
Ganti ID aset (ID setelah `assets/`) dalam topik dengan `a+`. Ini memilih properti kecepatan angin untuk semua aset turbin angin demo. Filter `+` topik menerima semua node dari satu tingkat dalam suatu topik. Topik Anda akan terlihat seperti contoh berikut.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-1111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

6. Masukkan pernyataan query aturan berikut. Ganti topik di FROM bagian dengan topik notifikasi Anda.

```
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-1111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-3333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

7. Di bawah Setel satu atau beberapa tindakan, pilih Tambah tindakan.



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. Pada halaman Pilih tindakan, pilih Pisahkan pesan menjadi beberapa kolom tabel DynamoDB (Dynamo). DBv2



Select an action

Select an action.

- ☐  Insert a message into a DynamoDB table
DYNAMODB
- ☒  Split message into multiple columns of a DynamoDB table (DynamoDBv2)
DYNAMODBv2
- ☐  Send a message to a Lambda function
LAMBDA

9. Pilih Konfigurasi tindakan di bagian bawah halaman.
10. Pada halaman Configure action, pilih Create a new resource.

Konsol DynamoDB terbuka di tab baru. Biarkan tab tindakan aturan tetap terbuka saat Anda menyelesaikan prosedur berikut.

Langkah 3: Buat tabel DynamoDB

Dalam prosedur ini, Anda membuat tabel Amazon DynamoDB untuk menerima data kecepatan angin dari tindakan aturan.

Untuk membuat tabel DynamoDB

1. Di dasbor konsol DynamoDB, pilih Buat tabel.
2. Masukkan nama untuk tabel Anda.

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.

i 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 Create

3. Untuk kunci Primer, lakukan hal berikut:
 - a. Masukkan **timestamp** sebagai kunci partisi.
 - b. Pilih jenis Angka.
 - c. Pilih kotak centang Add sort key.
 - d. Masukkan **asset** sebagai kunci sortir, dan biarkan tipe kunci sortir default String.
4. Pilih Buat.

Ketika Tabel sedang dibuat pemberitahuan menghilang, tabel Anda siap.

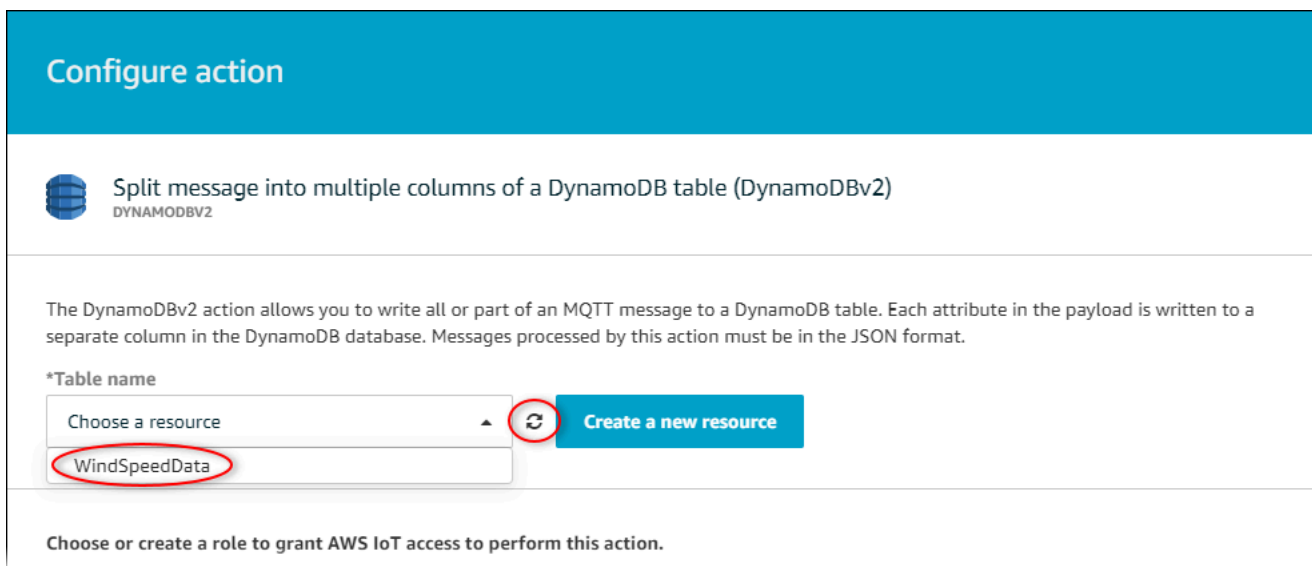
5. Kembali ke tab dengan halaman Configure action. Biarkan tab DynamoDB tetap terbuka saat Anda menyelesaikan prosedur berikut.

Langkah 4: Konfigurasi tindakan aturan DynamoDB

Dalam prosedur ini, Anda mengonfigurasi tindakan aturan Amazon DynamoDB untuk menyisipkan data dari pembaruan nilai properti ke tabel DynamoDB baru Anda.

Untuk mengkonfigurasi tindakan aturan DynamoDB

1. Pada halaman Configure action, refresh daftar nama Tabel, dan pilih tabel DynamoDB baru Anda.



Configure action

 Split message into multiple columns of a DynamoDB table (DynamoDBv2)

The DynamoDBv2 action allows you to write all or part of an MQTT message to a DynamoDB table. Each attribute in the payload is written to a separate column in the DynamoDB database. Messages processed by this action must be in the JSON format.

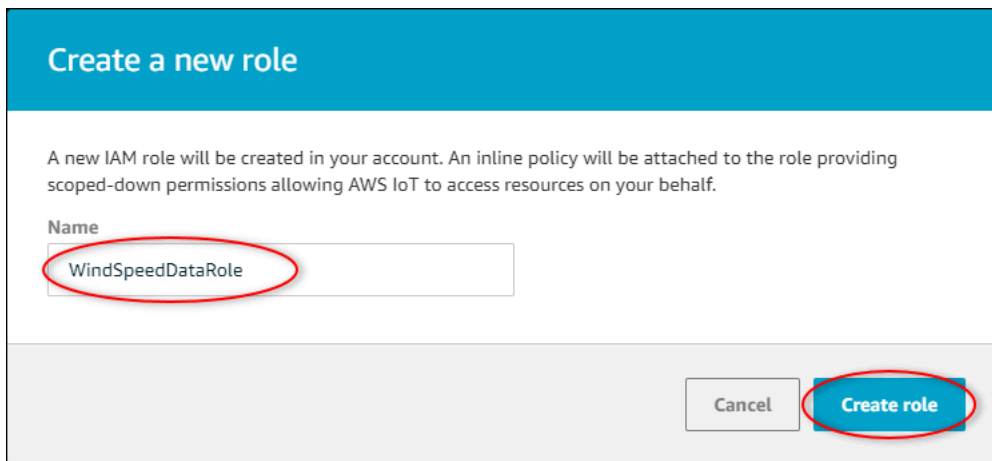
***Table name**

Choose a resource  **Create a new resource**

WindSpeedData

Choose or create a role to grant AWS IoT access to perform this action.

2. Pilih Buat peran untuk membuat peran IAM yang memberikan akses AWS IoT Core untuk melakukan tindakan aturan.
3. Masukkan nama peran dan pilih Buat peran.



4. Pilih Tambahkan tindakan.
5. Pilih Buat aturan di bagian bawah halaman untuk menyelesaikan pembuatan aturan.

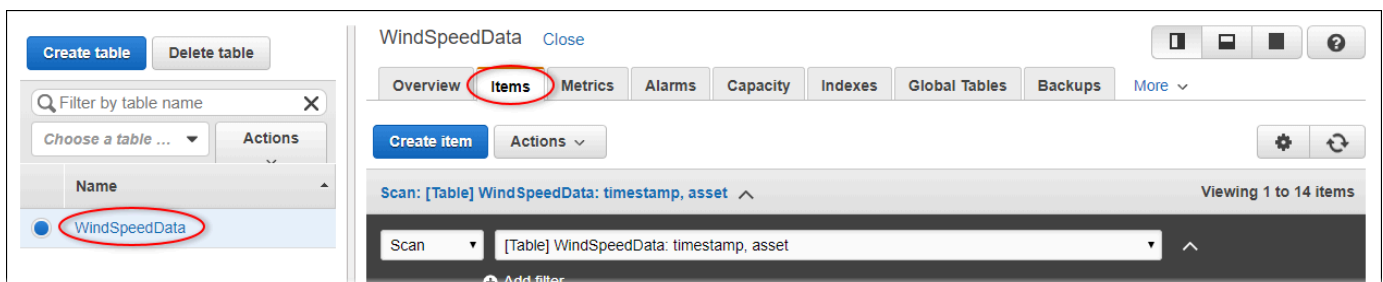
Data aset demo Anda akan mulai muncul di tabel DynamoDB Anda.

Langkah 5: Jelajahi data di DynamoDB

Dalam prosedur ini, Anda menjelajahi data kecepatan angin aset demo di tabel Amazon DynamoDB baru Anda.

Untuk mengeksplorasi data aset di DynamoDB

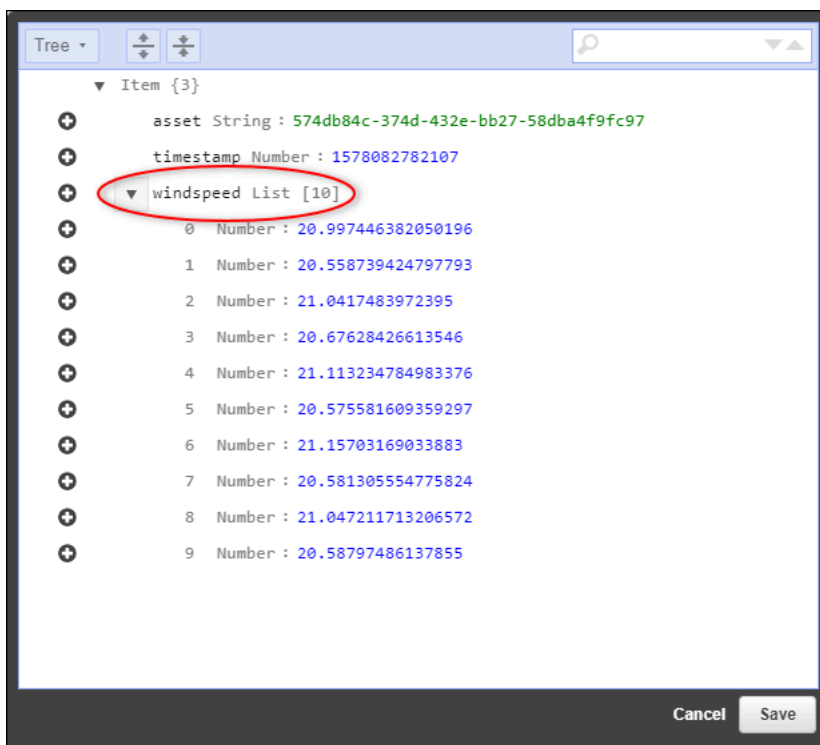
1. Kembali ke tab dengan tabel DynamoDB terbuka.
2. Dalam tabel yang Anda buat sebelumnya, pilih tab Item untuk melihat data dalam tabel. Segarkan halaman jika Anda tidak melihat baris dalam tabel. Jika baris tidak muncul setelah beberapa menit, lihat [Memecahkan masalah aturan \(DynamoDB\)](#).



3. Dalam satu baris dalam tabel, pilih ikon edit untuk memperluas data.

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

4. Pilih panah di sebelah windspeedstruktur untuk memperluas daftar titik data kecepatan angin. Setiap daftar mencerminkan sekumpulan titik data kecepatan angin yang dikirim AWS IoT SiteWise oleh demo ladang angin. Anda mungkin menginginkan format data yang berbeda jika Anda menyiapkan tindakan aturan untuk Anda gunakan sendiri. Untuk informasi selengkapnya, lihat [Kueri pemberitahuan properti aset di AWS IoT SiteWise](#).



Sekarang setelah Anda menyelesaikan tutorial, nonaktifkan atau hapus aturan dan hapus tabel DynamoDB Anda untuk menghindari biaya tambahan. Untuk membersihkan sumber daya Anda, lihat [Langkah 6: Bersihkan sumber daya setelah tutorial](#).

Langkah 6: Bersihkan sumber daya setelah tutorial

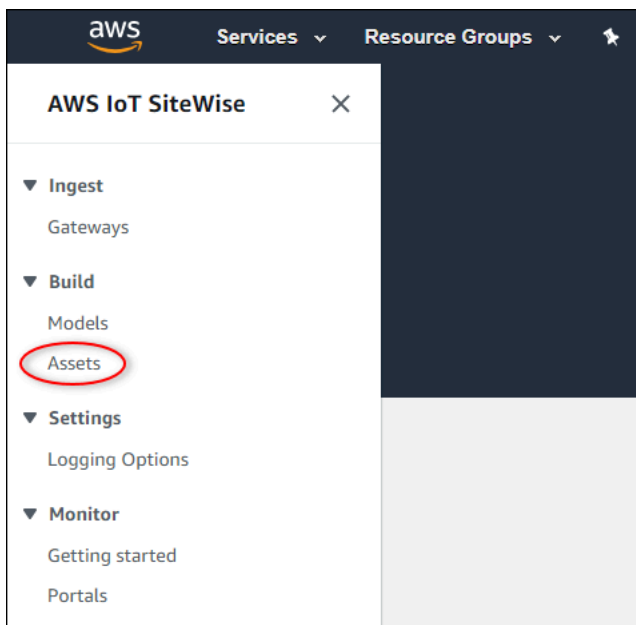
Setelah Anda menyelesaikan tutorial, bersihkan sumber daya Anda untuk menghindari biaya tambahan. Aset ladang angin demo Anda dihapus pada akhir durasi yang Anda pilih saat Anda

membuat demo. Anda juga dapat menghapus demo secara manual. Untuk informasi selengkapnya, lihat [Hapus AWS IoT SiteWise demo](#).

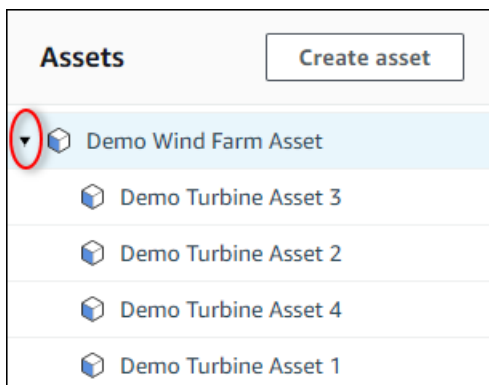
Gunakan prosedur berikut untuk menonaktifkan pemberitahuan pembaruan nilai properti (jika Anda tidak menghapus demo), menonaktifkan atau menghapus AWS IoT aturan Anda, dan menghapus tabel DynamoDB Anda.

Untuk menonaktifkan pemberitahuan pembaruan nilai properti pada properti aset

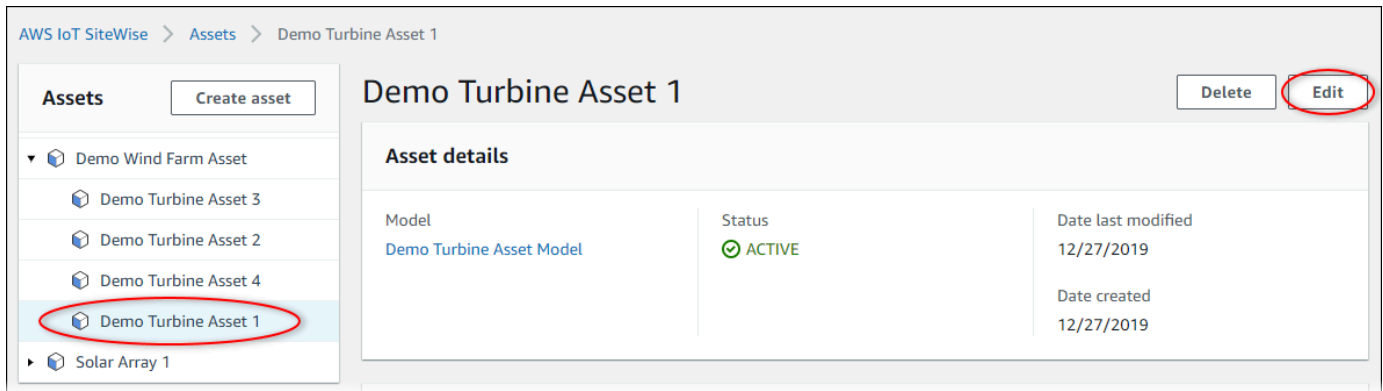
1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Aset.



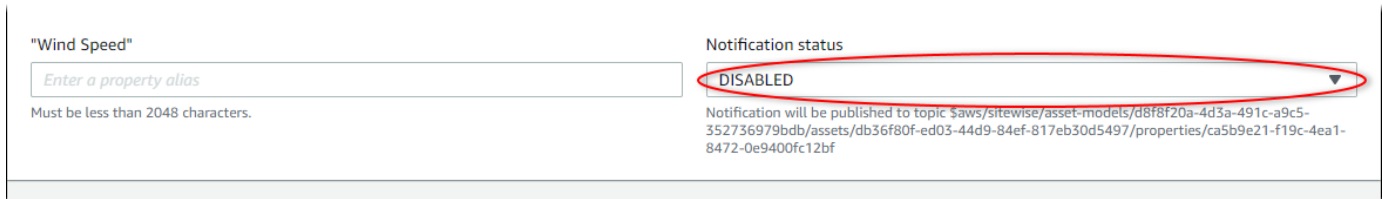
3. Pilih panah di sebelah Demo Wind Farm Asset untuk memperluas hierarki aset ladang angin.



4. Pilih turbin demo dan pilih Edit.



5. Perbarui Wind SpeedStatus Pemberitahuan properti ke DISABLED.

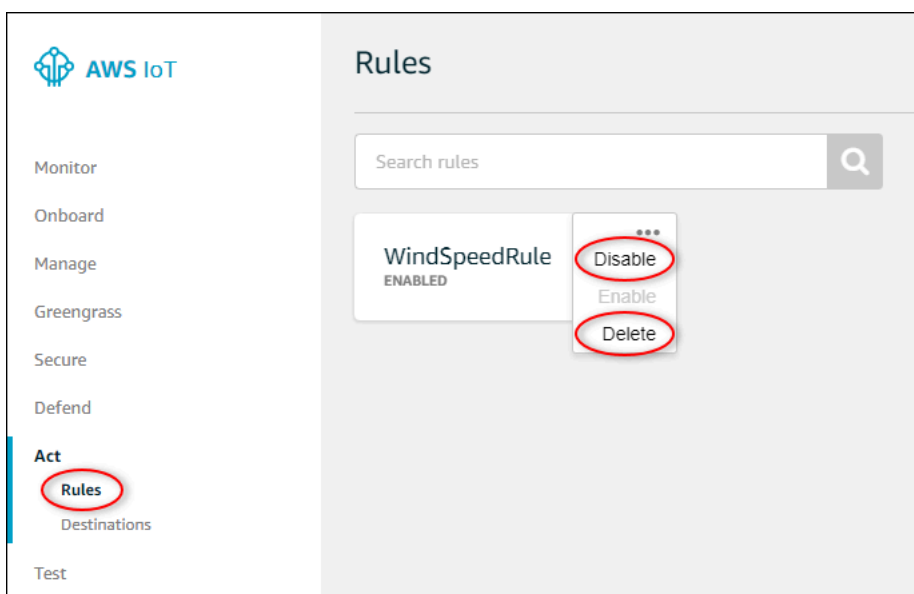


6. Pilih Simpan aset di bagian bawah halaman.

7. Ulangi langkah 4 hingga 6 untuk setiap aset turbin demo.

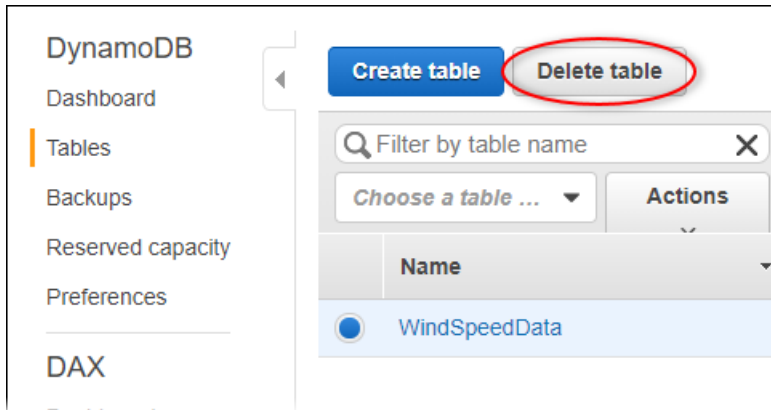
Untuk menonaktifkan atau menghapus aturan di AWS IoT Core

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi kiri, pilih Bertindak lalu pilih Aturan.
3. Pilih menu pada aturan Anda dan pilih Nonaktifkan atau Hapus.

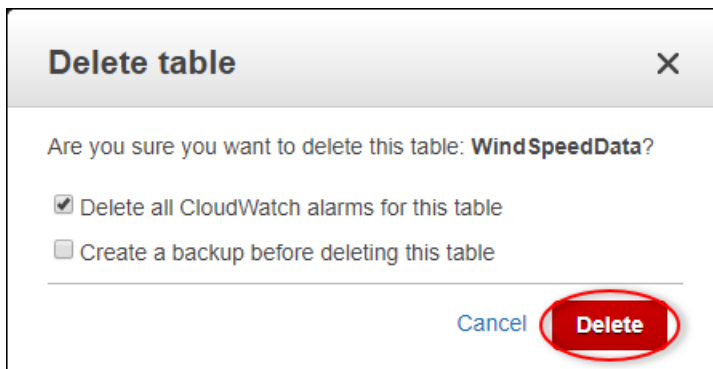


Untuk menghapus tabel DynamoDB

1. Arahkan ke konsol [DynamoDB](#).
2. Di panel navigasi kiri, pilih Tabel.
3. Pilih tabel yang Anda buat sebelumnya, WindSpeedData.
4. Pilih Hapus tabel.



5. Dalam dialog Hapus tabel, pilih Hapus.



Menelan data ke AWS IoT SiteWise

AWS IoT SiteWise dirancang untuk secara efisien mengumpulkan dan mengkorelasikan data industri dengan aset yang sesuai, mewakili berbagai aspek operasi industri. Dokumentasi ini berfokus pada aspek praktis dari menelan data ke dalam AWS IoT SiteWise, menawarkan beberapa metode yang disesuaikan dengan beragam kasus penggunaan industri. Untuk instruksi untuk membangun operasi industri virtual Anda, lihat [Model aset industri](#).

Anda dapat mengirim data industri untuk AWS IoT SiteWise menggunakan salah satu opsi berikut:

- AWS IoT SiteWise Edge —Gunakan [gateway SiteWise Edge](#) sebagai perantara antara AWS IoT SiteWise dan server data Anda. AWS IoT SiteWise menyediakan AWS IoT Greengrass komponen yang dapat Anda terapkan pada platform apa pun yang dapat dijalankan AWS IoT Greengrass untuk menyiapkan gateway SiteWise Edge. Opsi ini mendukung penautan dengan protokol server [OPC UA](#).
- AWS IoT SiteWise API —Gunakan [AWS IoT SiteWise API](#) untuk mengunggah data dari sumber lain. Gunakan [BatchPutAssetPropertyValue](#) API streaming kami untuk konsumsi dalam hitungan detik, atau [CreateBulkImportJob](#) API berorientasi batch untuk memfasilitasi konsumsi hemat biaya dalam batch yang lebih besar.
- AWS IoT Aturan inti —Gunakan [aturan AWS IoT Inti](#) untuk mengunggah data dari pesan MQTT yang diterbitkan oleh AWS IoT sesuatu atau layanan lain. AWS
- AWS IoT Events tindakan —Gunakan [AWS IoT Events tindakan](#) yang dipicu oleh peristiwa tertentu di AWS IoT Events. Metode ini cocok untuk skenario di mana unggahan data terkait dengan kejadian peristiwa.
- AWS IoT Greengrass manajer aliran —Gunakan [pengelola AWS IoT Greengrass aliran](#) untuk mengunggah data dari sumber data lokal menggunakan perangkat edge. Opsi ini melayani situasi di mana data berasal dari lokasi lokal atau tepi.

Metode ini menawarkan berbagai solusi untuk mengelola data dari berbagai sumber. Selami detail setiap opsi untuk mendapatkan pemahaman komprehensif tentang kemampuan konsumsi data yang disediakan. AWS IoT SiteWise

Mengelola aliran data untuk AWS IoT SiteWise

Aliran data adalah sumber daya yang berisi data deret waktu historis. Setiap aliran data diidentifikasi oleh alias unik, sehingga lebih mudah untuk melacak asal untuk setiap bagian data. Aliran data

secara otomatis dibuat AWS IoT SiteWise ketika data seri pertama kali diterima. Jika data seri pertama kali diidentifikasi dengan alias, AWS IoT SiteWise buat aliran data baru dengan alias tersebut, asalkan tidak ada properti aset yang sudah ditetapkan alias tersebut. Atau, jika data seri pertama kali diidentifikasi dengan ID aset dan ID properti, AWS IoT SiteWise buat aliran data baru dan kaitkan aliran data tersebut dengan properti aset.

Ada dua cara untuk menetapkan alias ke properti aset. Metode yang digunakan tergantung pada apakah data dikirim ke AWS IoT SiteWise pertama, atau aset dibuat terlebih dahulu.

- Jika data dikirim ke AWS IoT SiteWise pertama, ini secara otomatis membuat aliran data dengan alias yang ditetapkan. Saat aset dibuat nanti, gunakan [AssociateTimeSeriesToAssetProperty](#) API untuk mengaitkan aliran data dan aliasnya ke properti aset.
- Jika aset dibuat terlebih dahulu, gunakan [UpdateAssetProperty](#) API untuk menetapkan alias ke properti aset. Ketika data kemudian dikirim ke AWS IoT SiteWise, aliran data secara otomatis dibuat dan dikaitkan dengan properti aset.

Saat ini, Anda hanya dapat mengaitkan aliran data dengan pengukuran. Pengukuran adalah jenis properti aset yang mewakili aliran data sensor mentah perangkat, seperti nilai suhu stempel waktu atau nilai rotasi per menit (RPM) stempel waktu.

Ketika pengukuran ini menentukan metrik atau transformasi, data yang masuk memicu perhitungan tertentu. Penting untuk dicatat bahwa properti aset hanya dapat ditautkan ke satu aliran data pada satu waktu.

AWS IoT SiteWise menggunakan TimeSeries sumber daya Amazon Resource Name (ARN) untuk menentukan biaya penyimpanan Anda. Untuk informasi selengkapnya, silakan lihat [Harga AWS IoT SiteWise](#).

Bagian berikut menunjukkan cara menggunakan AWS IoT SiteWise konsol atau API untuk mengelola aliran data.

Topik

- [Konfigurasi izin dan pengaturan](#)
- [Kaitkan aliran data ke properti aset](#)
- [Memutuskan aliran data dari properti aset](#)
- [Menghapus aliran data](#)
- [Memperbarui alias properti aset](#)

- [Skenario umum](#)

Konfigurasi izin dan pengaturan

Aliran data secara otomatis dibuat AWS IoT SiteWise ketika data seri pertama kali diterima. Jika data yang dicerna tidak terkait dengan properti aset, AWS IoT SiteWise buat aliran data baru yang tidak terkait yang dapat dikonfigurasi untuk dikaitkan dengan properti aset. Konfigurasi kontrol akses gateway yang mengirim data ke AWS IoT SiteWise, menggunakan kebijakan IAM untuk menentukan jenis data yang akan dicerna.

Kebijakan IAM berikut menonaktifkan konsumsi data yang tidak terkait dari gateway, sambil tetap mengizinkan penyerapan data ke aliran data yang terkait dengan properti aset:

Example Kebijakan pengguna IAM yang menonaktifkan konsumsi data yang tidak terkait dari gateway

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAssetIdAndPropertyId",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*"
    },
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAliasWithAssociatedAssetProperty",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
      "Condition": {
        "StringLikeIfExists": {
          "iotsitewise:isAssociatedWithAssetProperty": "true"
        }
      }
    },
    {
      "Sid": "DenyPutAssetPropertyValuesUsingAliasWithNoAssociatedAssetProperty",
      "Effect": "Deny",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
      "Condition": {
```

```
        "StringLikeIfExists": {
            "iotsitewise:isAssociatedWithAssetProperty": "false"
        }
    }
}
]
```

Example Kebijakan pengguna IAM yang menonaktifkan semua konsumsi data dari gateway

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

Kaitkan aliran data ke properti aset

Kelola aliran data Anda menggunakan Konsol AWS IoT SiteWise atau AWS CLI.

Console

Gunakan AWS IoT SiteWise konsol untuk mengelola aliran data Anda.

Untuk mengelola aliran data (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aliran data.
3. Pilih aliran data dengan memfilter alias aliran data, atau memilih Aliran data yang tidak terkait di menu tarik-turun filter.
4. Pilih aliran data yang akan diperbarui. Anda dapat memilih beberapa aliran data. Klik Kelola aliran data di kanan atas.

5. Pilih aliran data yang akan dikaitkan dari Perbarui asosiasi aliran data, dan klik Pilih pengukuran tombol.
6. Di bagian Pilih pengukuran, temukan properti pengukuran aset yang sesuai. Pilih pengukuran lalu klik Pilih.
7. Lakukan langkah 4 dan 5 untuk aliran data lain yang dipilih pada langkah 3. Tetapkan properti aset ke semua aliran data.
8. Pilih Perbarui untuk melakukan perubahan. Spanduk konfirmasi yang berhasil ditampilkan untuk mengonfirmasi pembaruan.

AWS CLI

Untuk mengaitkan aliran data (diidentifikasi dengan aliasnya) ke properti aset (diidentifikasi olehnya IDs), jalankan perintah berikut:

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

Memutuskan aliran data dari properti aset

Console

Gunakan AWS IoT SiteWise konsol untuk memisahkan aliran data Anda dari properti aset.

Untuk memisahkan aliran data dari properti aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aliran data.
3. Pilih aliran data dengan memfilter alias aliran data, atau memilih Aliran data terkait di menu tarik-turun filter.
4. Pilih aliran data untuk memisahkan. Kolom Alias data stream harus berisi alias. Kolom nama aset dan nama properti Aset harus berisi nilai properti aset yang terkait dengan aliran data. Anda dapat memilih beberapa aliran data.
5. Klik Kelola aliran data di kanan atas.

6. Di bagian Perbarui asosiasi aliran data, klik X di kolom Nama pengukuran. `submittedStatus` akan muncul di kolom Status.
7. Pilih Perbarui untuk melakukan perubahan. Aliran data sekarang dipisahkan dari properti aset, dan alias sekarang digunakan untuk mengidentifikasi aliran data.

AWS CLI

Untuk memisahkan aliran data dari properti aset, (diidentifikasi oleh ID s dan aliasnya), jalankan perintah berikut:

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Aliran data sekarang dipisahkan dari properti aset, dan alias digunakan untuk mengidentifikasi aliran data. Alias tidak lagi terkait dengan properti aset, karena sekarang dikaitkan dengan aliran data.

Menghapus aliran data

Ketika properti dihapus dari model aset, AWS IoT SiteWise menghapus properti dan aliran datanya dari semua aset yang dikelola oleh model aset. Ini juga menghapus semua properti dan aliran data aset mereka ketika aset dihapus. Jika data stream harus dipertahankan, data harus dipisahkan dari properti aset sebelum dihapus.

Warning

Ketika properti dihapus dari aset, aliran data terkait juga dihapus. Untuk mempertahankan aliran data, lepaskan dari properti aset terlebih dahulu, sebelum menghapus properti dari model aset, atau menghapus aset.

Console

Gunakan AWS IoT SiteWise konsol untuk memisahkan aliran data Anda dari properti aset.

Untuk menghapus aliran data (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aliran data.
3. Pilih aliran data dengan memfilter alias aliran data.
4. Pilih aliran data yang akan dihapus. Anda dapat memilih beberapa aliran data.
5. Pilih tombol Hapus untuk menghapus aliran data.

AWS CLI

Gunakan [DeleteTimeSeries](#) API untuk menghapus aliran data tertentu, dengan aliasnya.

```
aws iotsitewise delete-time-series \  
  --alias <data-stream-alias>
```

Memperbarui alias properti aset

Alias harus unik dalam suatu AWS wilayah. Ini termasuk alias properti aset dan aliran data. Jangan menetapkan alias ke properti aset, jika properti atau aliran data lain menggunakan alias tersebut.

Console

Gunakan AWS IoT SiteWise konsol untuk memperbarui alias properti aset.

Untuk memperbarui alias properti aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset dari tabel.
4. Klik tombol Edit.
5. Pilih tipe Properti di tabel Properties.
6. Temukan properti, dan ketik alias baru di bidang teks alias properti.
7. Klik tombol Simpan untuk menyimpan perubahan.

AWS CLI

Untuk memperbarui alias pada properti aset, jalankan perintah berikut:

```
aws iotsitewise update-asset-property \  
  --asset-id <asset-ID> \  
  --property-id <property-ID> \  
  --property-alias <asset-property-alias> \  
  --property-notification-state <ENABLED|DISABLED>
```

Note

Jika pemberitahuan properti saat ini diaktifkan, itu harus disediakan lagi untuk memastikannya terus diaktifkan.

Skenario umum

Memindahkan aliran data

Untuk mengubah asosiasi aliran data ke properti aset lain, pertama-tama lepaskan aliran data dari properti aset saat ini. Saat memisahkan aliran data dari properti aset, harus ada alias yang ditetapkan ke properti aset tersebut.

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Sekarang tetapkan kembali aliran data ke properti aset baru.

```
aws iotsitewise associate-time-series-from-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <new-asset-ID> \  
  --propertyId <new-property-ID>
```

Kesalahan saat menetapkan alias ke properti aset

Saat menggunakan UpdateAssetProperty API untuk menetapkan alias ke properti, Anda mungkin melihat pesan galat berikut:

```
Given alias <data-stream-alias> for property <property-name> with ID <property-ID>
already in use by another property or data stream
```

Pesan kesalahan ini menunjukkan alias tidak ditetapkan ke properti, karena saat ini digunakan oleh properti lain atau aliran data.

Ini terjadi jika data sedang dicerna AWS IoT SiteWise dengan alias. Ketika data dikirim dengan alias yang tidak digunakan oleh aliran data lain atau properti aset, aliran data baru dibuat dengan alias itu. Dua opsi di bawah ini menyelesaikan masalah.

- Gunakan AssociateTimeSeriesToAssetProperty API untuk mengaitkan aliran data dengan aliasnya ke properti aset.
- Hentikan sementara konsumsi data dan hapus aliran data. Gunakan UpdateAssetProperty API untuk menetapkan alias ke properti aset, lalu aktifkan kembali konsumsi data.

Kesalahan saat mengaitkan aliran data ke properti aset

Saat mengaitkan aliran data ke properti aset, pesan galat berikut akan terlihat.

```
assetProperty <property-name> with assetId <asset-ID> propertyId <property-ID> contains
data
```

Pesan galat ini menunjukkan properti aset sudah dikaitkan dengan aliran data yang berisi data. Aliran data tersebut harus dipisahkan atau dihapus, sebelum mengaitkan aliran data lain ke properti aset tersebut.

Note

Saat memisahkan aliran data dari properti aset, alias yang ditetapkan ke properti diberikan ke aliran data. Agar alias tersebut tetap ditetapkan ke properti, tetapkan alias baru ke properti tersebut sebelum memisahkan aliran data.

Untuk melestarikan data yang disimpan di properti aset lakukan hal berikut:

- Pastikan tidak ada data yang dicerna ke properti aset, untuk mencegah pembuatan aliran data baru.
- Gunakan `UpdateAssetProperty` API untuk menyetel alias baru yang diberikan ke aliran data yang ditetapkan saat ini.
- Gunakan `DisassociateTimeSeriesFromAssetProperty` API untuk memisahkan aliran data saat ini dari properti aset.
- Gunakan `AssociateTimeSeriesToAssetProperty` API untuk mengaitkan aliran data yang diinginkan ke properti aset.

Jika data yang disimpan di properti aset harus dihapus, lakukan hal berikut:

- Pastikan tidak ada data yang dicerna ke properti aset, untuk mencegah pembuatan aliran data baru.
- Gunakan `DeleteTimeSeries` API untuk menghapus aliran data yang ditetapkan saat ini.
- Gunakan `AssociateTimeSeriesToAssetProperty` API untuk mengaitkan aliran data yang diinginkan ke properti aset.

Menelan data dengan AWS IoT SiteWise APIs

Gunakan AWS IoT SiteWise APIs untuk mengirim data industri stempel waktu ke atribut dan properti pengukuran aset Anda. APIs Menerima struktur muatan yang mengandung timestamp-quality-value (TQV).

BatchPutAssetPropertyValue API

Gunakan [BatchPutAssetPropertyValue](#) operasi untuk mengunggah data Anda. Dengan operasi ini, Anda dapat mengunggah beberapa entri data sekaligus untuk mengumpulkan data dari beberapa perangkat dan mengirimkan semuanya dalam satu permintaan.

Important

[BatchPutAssetPropertyValue](#) Operasi tunduk pada kuota berikut:

- Hingga 10 [entri](#) per permintaan.
- Hingga 10 [nilai properti](#) (titik data TQV) per entri.

- AWS IoT SiteWise menolak data apa pun dengan stempel waktu bertanggal lebih dari 7 hari di masa lalu atau lebih dari 10 menit di masa depan.

Untuk informasi selengkapnya tentang kuota ini, lihat [BatchPutAssetPropertyValuedi Referensi AWS IoT SiteWise API](#).

Untuk mengidentifikasi properti aset, tentukan salah satu dari berikut ini:

- Properti `propertyId` aset yang dikirimkan ke data. `assetId`
- `ThepropertyAlias`, yang merupakan alias aliran data (misalnya, `/company/windfarm/3/turbine/7/temperature`). Untuk menggunakan opsi ini, Anda harus terlebih dahulu menetapkan alias properti aset Anda. Untuk mengatur alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Contoh berikut menunjukkan cara mengirim suhu turbin angin dan rotasi per menit (RPM) pembacaan dari muatan yang disimpan dalam file JSON.

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

Contoh payload di `batch-put-payload.json` memiliki konten berikut.

```
{
  "enablePartialEntryProcessing": true,
  "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"
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
      "propertyValues": [
        {
          "value": {
            "nullValue": {"valueType": "D"}
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          },
          "quality": "BAD"
        }
      ]
    }
  ]
}

```

Menentukan `enablePartialEntryProcessing` sebagai `true` memungkinkan konsumsi semua nilai yang tidak mengakibatkan kegagalan. Perilaku bawaannya adalah `false`. Jika nilai tidak valid, seluruh entri gagal tertelan.

Setiap entri dalam payload berisi `entryId` yang dapat Anda definisikan sebagai string unik apa pun. Jika ada entri permintaan yang gagal, setiap kesalahan akan berisi permintaan `entryId` yang sesuai sehingga Anda tahu permintaan mana yang harus dicoba lagi.

Setiap struktur dalam daftar `propertyValues` adalah struktur timestamp-quality-value (TQV) yang berisi `value`, `timestamp`, dan opsional `quality`.

- `value`— Struktur yang berisi salah satu bidang berikut, tergantung pada jenis properti yang ditetapkan:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `nullValue`— Struktur dengan bidang berikut yang menunjukkan jenis nilai properti dengan nilai Null dan kualitas atau. BAD UNCERTAIN
 - `valueType`— Enum dari {"B", "D", "S", "I"}
- `timestamp`— Struktur yang berisi waktu epoch Unix saat ini dalam hitungan detik,. `timeInSeconds` Anda juga dapat mengatur `offsetInNanos` kunci dalam `timestamp` struktur jika Anda memiliki data yang tepat untuk sementara. AWS IoT SiteWise menolak setiap titik data dengan stempel waktu yang lebih tua dari 7 hari di masa lalu atau lebih baru dari 10 menit di masa depan.
- `quality`— (Opsional) Salah satu string kualitas berikut:
 - GOOD— (Default) Data tidak terpengaruh oleh masalah apa pun.
 - BAD— Data dipengaruhi oleh masalah seperti kegagalan sensor.
 - UNCERTAIN— Data dipengaruhi oleh masalah seperti ketidakakuratan sensor.

Untuk informasi selengkapnya tentang cara AWS IoT SiteWise menangani kualitas data dalam komputasi, lihat [Kualitas data dalam ekspresi rumus](#).

CreateBulkImportJob API

Gunakan `CreateBulkImportJob` API untuk mengimpor data dalam jumlah besar dari Amazon S3. Data Anda harus disimpan dalam format CSV di Amazon S3. File data dapat memiliki kolom berikut.

Note

Data yang lebih lama dari 1 Januari 1970 00:00:00 UTC tidak didukung.
Untuk mengidentifikasi properti aset, tentukan salah satu dari berikut ini.

- Properti `PROPERTY_ID` aset yang Anda kirim datanya. `ASSET_ID`
- `THEALIAS`, yang merupakan alias aliran data (misalnya, `/company/windfarm/3/turbine/7/temperature`). Untuk menggunakan opsi ini, Anda harus terlebih dahulu menetapkan alias properti aset Anda. Untuk mempelajari cara menyetel alias properti, lihat [the section called “Kelola aliran data”](#).

- `ALIAS`— Alias yang mengidentifikasi properti, seperti jalur aliran data server OPC UA (misalnya, `/company/windfarm/3/turbine/7/temperature`). Untuk informasi selengkapnya, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).
- `ASSET_ID`— ID aset.
- `PROPERTY_ID`— ID properti aset.
- `DATA_TYPE`— Jenis data properti dapat menjadi salah satu dari berikut ini.
 - `STRING`— Sebuah string dengan hingga 1024 byte.
 - `INTEGER`— Bilangan bulat 32-bit yang ditandatangani dengan rentang $[-2,147.483.648, 2.147.483.647]$.
 - `DOUBLE`— Nomor floating point dengan jangkauan $[-10^{100}, 10^{100}]$ dan presisi ganda IEEE 754.
 - `BOOLEAN`— `true` atau `false`.
- `TIMESTAMP_SECONDS`— Stempel waktu titik data, dalam waktu zaman Unix.
- `TIMESTAMP_NANO_OFFSET`— Offset nanodetik tertutup dari `TIMESTAMP_SECONDS`
- `QUALITY`— (Opsional) Kualitas nilai properti aset. Nilainya bisa menjadi salah satu dari berikut ini.
 - `GOOD`— (Default) Data tidak terpengaruh oleh masalah apa pun.
 - `BAD`— Data dipengaruhi oleh masalah seperti kegagalan sensor.
 - `UNCERTAIN`— Data dipengaruhi oleh masalah seperti ketidakakuratan sensor.

Untuk informasi selengkapnya tentang cara AWS IoT SiteWise menangani kualitas data dalam komputasi, lihat [Kualitas data dalam ekspresi rumus](#).

- `VALUE`— Nilai properti aset.

Example file data dalam format.csv

```
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 menyediakan operasi API berikut untuk membuat pekerjaan impor massal dan mendapatkan informasi tentang pekerjaan yang ada.

- [CreateBulkImportJob](#)— Membuat pekerjaan impor massal baru.
- [DescribeBulkImportJob](#)— Mengambil informasi tentang pekerjaan impor massal.
- [ListBulkImportJob](#)— Mengambil daftar ringkasan halaman dari semua pekerjaan impor massal.

Buat pekerjaan impor AWS IoT SiteWise massal (AWS CLI)

Gunakan operasi [CreateBulkImportJob](#) API untuk mentransfer data dari Amazon S3 ke. AWS IoT SiteWise [CreateBulkImportJob](#) API memungkinkan konsumsi volume besar data historis, dan buffer konsumsi aliran data analitik dalam batch kecil. Ini memberikan primitif hemat biaya untuk konsumsi data. Contoh berikut menggunakan AWS CLI.

Important

Sebelum membuat pekerjaan impor massal, Anda harus mengaktifkan tingkat AWS IoT SiteWise hangat atau tingkat AWS IoT SiteWise dingin. Untuk informasi selengkapnya, lihat [Konfigurasi pengaturan penyimpanan di AWS IoT SiteWise](#).

[CreateBulkImportJob](#) API mendukung konsumsi data historis ke dalam AWS IoT SiteWise dengan opsi untuk mengatur parameter. `adaptive-ingestion-flag`

- Saat disetel ke `false`, API akan menyerap data historis tanpa memicu perhitungan atau notifikasi.

- Saat disetel ke `true`, API akan menyerap data baru, menghitung metrik, dan mengubah data untuk mengoptimalkan analitik dan notifikasi yang sedang berlangsung dalam tujuh hari.

Jalankan perintah berikut. Ganti `file-name` dengan nama file yang berisi konfigurasi pekerjaan impor massal.

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

Example Konfigurasi pekerjaan impor massal

Berikut ini adalah contoh pengaturan konfigurasi:

- Ganti `adaptive-ingestion-flag` dengan `true` atau `false`.
 - Jika disetel ke `false`, pekerjaan impor massal akan memasukkan data historis ke dalam AWS IoT SiteWise.
 - Jika disetel ke `true`, pekerjaan impor massal melakukan hal berikut:
 - Menyerap data baru ke dalam AWS IoT SiteWise.
 - Menghitung metrik dan transformasi, dan mendukung pemberitahuan untuk data dengan cap waktu yang dalam tujuh hari.
- Ganti `delete-files-after-import-flag` dengan `true` untuk menghapus data dari bucket data Amazon S3 setelah masuk ke penyimpanan tingkat AWS IoT SiteWise hangat.
- Ganti `amzn-s3-demo-bucket` dengan `-for-errors` nama bucket Amazon S3 tempat kesalahan yang terkait dengan pekerjaan impor massal ini dikirim.
- Ganti `amzn-s3-demo-bucket` dengan `-for-errors-prefix` awalan bucket Amazon S3 tempat kesalahan yang terkait dengan pekerjaan impor massal ini dikirim.

Amazon S3 menggunakan awalan sebagai nama folder untuk mengatur data dalam bucket. Setiap objek Amazon S3 memiliki kunci yang merupakan pengenalan uniknya di bucket. Setiap objek dalam bucket memiliki satu kunci. Prefiks harus diakhiri dengan garis miring ke depan (/). Untuk informasi selengkapnya, lihat [Mengatur objek menggunakan awalan](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

- Ganti `amzn-s3-demo-bucket` dengan `-data` nama bucket Amazon S3 tempat datanya diimpor.
- Ganti `data-bucket-key` dengan kunci objek Amazon S3 yang berisi data Anda. Setiap objek memiliki kunci yang berupa pengidentifikasi unik. Setiap objek memiliki tepat satu kunci.

- Ganti *data-bucket-version-id* dengan ID versi untuk mengidentifikasi versi tertentu dari objek Amazon S3 yang berisi data Anda. Parameter ini bersifat opsional.
- Ganti *column-name* dengan nama kolom yang ditentukan dalam file.csv.
- Ganti *job-name* dengan nama unik yang mengidentifikasi pekerjaan impor massal.
- Ganti *job-role-arn* dengan peran IAM yang memungkinkan AWS IoT SiteWise untuk membaca data Amazon S3.

Note

Pastikan peran Anda memiliki izin yang ditunjukkan dalam contoh berikut. Ganti *amzn-s3-demo-bucket* dengan *-data* nama bucket Amazon S3 yang berisi data Anda. Selain itu, ganti *amzn-s3-demo-bucket-for-errors* dengan nama bucket Amazon S3 tempat kesalahan yang terkait dengan pekerjaan impor massal ini dikirim.

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

```
]
}
```

```
{
  "adaptiveIngestion": adaptive-ingestion-flag,
  "deleteFilesAfterImport": delete-files-after-import-flag,
  "errorReportLocation": {
    "bucket": "amzn-s3-demo-bucket-for-errors",
    "prefix": "amzn-s3-demo-bucket-for-errors-prefix"
  },
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket-data",
      "key": "data-bucket-key",
      "versionId": "data-bucket-version-id"
    }
  ],
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [ "column-name" ]
      }
    }
  },
  "jobName": "job-name",
  "jobRoleArn": "job-role-arn"
}
```

Example response

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

Jelaskan pekerjaan impor AWS IoT SiteWise massal (AWS CLI)

Gunakan operasi [DescribeBulkImportJob](#) API untuk mengambil informasi tentang pekerjaan impor massal tertentu di AWS IoT SiteWise. Operasi ini mengembalikan detail seperti status

pekerjaan, waktu pembuatan, dan informasi kesalahan jika pekerjaan gagal. Anda dapat menggunakan operasi ini untuk memantau kemajuan pekerjaan dan memecahkan masalah. Untuk menggunakannya `DescribeBulkImportJob`, Anda memerlukan ID pekerjaan dari `CreateBulkImportJob` operasi. API mengembalikan informasi berikut:

- Daftar file yang diimpor, termasuk lokasi dan kunci bucket Amazon S3
- Lokasi laporan kesalahan (jika ada)
- Detail konfigurasi pekerjaan, seperti format file dan nama kolom CSV
- Pembuatan Job dan stempel waktu pembaruan terakhir
- Status pekerjaan saat ini (misalnya, apakah pekerjaan sedang berlangsung, selesai, atau gagal)
- Peran IAM ARN digunakan untuk pekerjaan impor

Untuk pekerjaan yang telah selesai, tinjau hasilnya untuk mengonfirmasi integrasi data yang berhasil. Jika pekerjaan gagal, periksa detail kesalahan untuk mendiagnosis dan menyelesaikan masalah.

Ganti *job-ID* dengan ID pekerjaan impor massal yang ingin Anda ambil.

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

Example response

```
{
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket1",
      "key": "100Tags12Hours.csv"
    },
    {
      "bucket": "amzn-s3-demo-bucket2",
      "key": "BulkImportData1MB.csv"
    },
    {
      "bucket": " amzn-s3-demo-bucket3",
      "key": "UnmodeledBulkImportData1MB.csv"
    }
  ],
  "errorReportLocation": {
    "prefix": "errors/",
    "bucket": "amzn-s3-demo-bucket-for-errors"
  }
}
```

```

},
"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"
}

```

Daftar pekerjaan impor AWS IoT SiteWise massal (AWS CLI)

Gunakan operasi [ListBulkImportJobs](#) API untuk mengambil daftar ringkasan untuk pekerjaan impor massal di AWS IoT SiteWise. Operasi ini menyediakan cara yang efisien untuk memantau dan mengelola proses impor data Anda. Ini mengembalikan informasi kunci berikut untuk setiap pekerjaan:

- ID Pekerjaan. Pengidentifikasi unik untuk setiap pekerjaan impor massal
- Nama Job. Nama yang Anda tetapkan untuk pekerjaan saat membuatnya
- Status saat ini. Status pekerjaan saat ini (misalnya, SELESAI, BERJALAN, GAGAL)

[ListBulkImportJobs](#) sangat berguna untuk mendapatkan gambaran menyeluruh dari semua pekerjaan impor massal Anda. Ini dapat membantu Anda melacak beberapa impor data, mengidentifikasi pekerjaan apa pun yang memerlukan perhatian, dan mempertahankan alur kerja yang terorganisir. Operasi ini mendukung pagination, memungkinkan Anda untuk mengambil sejumlah besar ringkasan pekerjaan secara efisien. Anda dapat menggunakan pekerjaan yang IDs dikembalikan oleh operasi ini dengan [DescribeBulkImportJob](#) operasi untuk mengambil informasi lebih rinci tentang pekerjaan

tertentu. Proses dua langkah ini memungkinkan Anda untuk terlebih dahulu mendapatkan pandangan tingkat tinggi dari semua pekerjaan, dan kemudian menelusuri detail pekerjaan yang diminati. Saat menggunakan `ListBulkImportJobs`, Anda dapat menerapkan filter untuk mempersempit hasil. Misalnya, Anda dapat memfilter pekerjaan berdasarkan statusnya untuk mengambil hanya pekerjaan yang telah selesai atau hanya menjalankan pekerjaan. Fitur ini membantu Anda fokus pada informasi yang paling relevan untuk tugas Anda saat ini. Operasi juga mengembalikan a `nextToken` jika ada lebih banyak hasil yang tersedia. Anda dapat menggunakan token ini dalam panggilan berikutnya untuk mengambil kumpulan ringkasan pekerjaan berikutnya, memungkinkan Anda untuk mengulangi semua pekerjaan impor massal Anda bahkan jika Anda memiliki sejumlah besar dari mereka. Contoh berikut menunjukkan bagaimana menggunakan `ListBulkImportJobs` dengan AWS CLI untuk mengambil daftar pekerjaan selesai.

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

Example Respon untuk filter pekerjaan yang telah selesai

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

Perintah ini menunjukkan bagaimana menggunakan `ListBulkImportJobs` untuk mengambil daftar pekerjaan yang selesai dengan kegagalan. Maksimum diatur ke 50 hasil dan kami menggunakan token berikutnya untuk hasil paginasi.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED_WITH_FAILURES --max-results 50
--next-token "string"
```


Menelan data untuk AWS IoT SiteWise menggunakan aturan AWS IoT Core

Kirim data ke AWS IoT SiteWise dari AWS IoT hal-hal dan AWS layanan lainnya dengan menggunakan aturan di AWS IoT Core. Aturan mengubah pesan MQTT dan melakukan tindakan untuk berinteraksi dengan layanan. AWS Tindakan AWS IoT SiteWise aturan meneruskan data pesan ke [BatchPutAssetPropertyValue](#) operasi dari AWS IoT SiteWise API. Untuk informasi selengkapnya, lihat [Aturan](#) dan [AWS IoT SiteWise tindakan](#) di Panduan AWS IoT Pengembang.

Untuk mengikuti tutorial yang berjalan melalui langkah-langkah yang diperlukan untuk mengatur aturan yang menyerap data melalui bayangan perangkat, lihat [Menelan data AWS IoT SiteWise dari AWS IoT berbagai hal](#).

Anda juga dapat mengirim data dari AWS IoT SiteWise ke AWS layanan lain. Untuk informasi selengkapnya, lihat [Berinteraksi dengan AWS layanan lain](#).

Topik

- [Berikan AWS IoT akses yang diperlukan](#)
- [Konfigurasi tindakan AWS IoT SiteWise aturan](#)
- [Kurangi biaya dengan Basic Ingest di AWS IoT SiteWise](#)

Berikan AWS IoT akses yang diperlukan

Anda menggunakan peran IAM untuk mengontrol AWS sumber daya yang dapat diakses oleh setiap aturan. Sebelum membuat aturan, Anda harus membuat peran IAM dengan kebijakan yang memungkinkan aturan melakukan tindakan pada AWS sumber daya yang diperlukan. AWS IoT mengasumsikan peran ini saat menjalankan aturan.

Jika Anda membuat tindakan aturan di AWS IoT konsol, Anda dapat memilih aset root untuk membuat peran yang memiliki akses ke hierarki aset yang dipilih. Untuk informasi selengkapnya tentang cara menentukan peran aturan secara manual, lihat [Memberikan akses AWS IoT yang diperlukan](#) dan [izin peran Lulus di Panduan AWS IoT](#) Pengembang.

Untuk tindakan AWS IoT SiteWise aturan, Anda harus menentukan peran yang memungkinkan `iotsitewise:BatchPutAssetPropertyValue` akses ke properti aset tempat aturan mengirimkan data. Untuk meningkatkan keamanan, Anda dapat menentukan jalur hierarki AWS IoT SiteWise aset di `Condition` properti.

Contoh kebijakan kepercayaan berikut memungkinkan akses ke aset tertentu dan anak-anaknya.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/root node asset ID",
            "/root node asset ID/*"
          ]
        }
      }
    }
  ]
}
```

Hapus Condition dari kebijakan untuk mengizinkan akses ke semua aset Anda. Contoh kebijakan kepercayaan berikut memungkinkan akses ke semua aset Anda di Wilayah saat ini.

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

Konfigurasi tindakan AWS IoT SiteWise aturan

Tindakan AWS IoT SiteWise aturan mengirimkan data dari pesan MQTT yang memulai aturan ke properti aset di AWS IoT SiteWise Anda dapat mengunggah beberapa entri data ke properti aset yang berbeda secara bersamaan, untuk mengirim pembaruan untuk semua sensor perangkat dalam satu pesan. Anda juga dapat mengunggah beberapa titik data sekaligus untuk setiap entri data.

 Note

Saat Anda mengirim data AWS IoT SiteWise dengan tindakan aturan, data Anda harus memenuhi semua persyaratan `BatchPutAssetPropertyValue` operasi. Misalnya, data Anda tidak dapat memiliki stempel waktu lebih awal dari 7 hari dari waktu epoch Unix saat ini. Untuk informasi selengkapnya, lihat [Menyerap data dengan AWS IoT SiteWise API](#).

Untuk setiap entri data dalam tindakan aturan, Anda mengidentifikasi properti aset dan menentukan stempel waktu, kualitas, dan nilai setiap titik data untuk properti aset tersebut. Tindakan aturan mengharapkan string untuk semua parameter.

Untuk mengidentifikasi properti aset dalam entri, tentukan salah satu dari berikut ini:

- Asset ID (`assetId`) dan Property ID (`propertyId`) dari properti aset yang Anda kirim data. Anda dapat menemukan ID Aset dan ID Properti menggunakan file Konsol AWS IoT SiteWise. Jika Anda mengetahui ID Aset, Anda dapat menggunakan AWS CLI to call [DescribeAsset](#) untuk menemukan ID Properti.
- Alias properti (`propertyAlias`), yang merupakan alias aliran data (misalnya, `/company/windfarm/3/turbine/7/temperature`). Untuk menggunakan opsi ini, Anda harus terlebih dahulu menetapkan alias properti aset Anda. Untuk mempelajari cara menyetel alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Untuk stempel waktu di setiap entri, gunakan stempel waktu yang dilaporkan oleh peralatan Anda atau stempel waktu yang disediakan oleh. AWS IoT Core Stempel waktu memiliki dua parameter:

- Waktu dalam detik (`timeInSeconds`) — Waktu epoch Unix, dalam detik, di mana sensor atau peralatan melaporkan data.
- Offset dalam nanos (`offsetInNanos`) — (Opsional) Nanodetik offset dari waktu dalam detik.

 Important

Jika stempel waktu Anda adalah string, memiliki bagian desimal, atau tidak dalam hitungan detik, AWS IoT SiteWise tolak permintaan tersebut. Anda harus mengubah stempel waktu menjadi detik dan offset nanodetik. Gunakan fitur mesin AWS IoT aturan untuk mengonversi stempel waktu. Untuk informasi selengkapnya, lihat berikut ini:

- [Mendapatkan stempel waktu untuk perangkat yang tidak melaporkan waktu yang akurat](#)
- [Mengonversi stempel waktu yang dalam format string](#)

Anda dapat menggunakan templat substitusi untuk beberapa parameter dalam tindakan untuk melakukan perhitungan, memanggil fungsi, dan menarik nilai dari muatan pesan. Untuk informasi selengkapnya, lihat [Templat substitusi](#) di Panduan AWS IoT Pengembang.

Note

Karena ekspresi dalam templat substitusi dievaluasi secara terpisah dari SELECT pernyataan, Anda tidak dapat menggunakan templat substitusi untuk mereferensikan alias yang dibuat menggunakan klausa. AS Anda hanya dapat mereferensikan informasi yang ada di muatan asli, selain fungsi dan operator yang didukung.

Topik

- [Mendapatkan stempel waktu untuk perangkat yang tidak melaporkan waktu yang akurat](#)
- [Mengonversi stempel waktu yang dalam format string](#)
- [Mengonversi string stempel waktu presisi nanodetik](#)
- [Contoh konfigurasi aturan](#)
- [Memecahkan masalah tindakan aturan](#)

Mendapatkan stempel waktu untuk perangkat yang tidak melaporkan waktu yang akurat

Jika sensor atau peralatan Anda tidak melaporkan data waktu yang akurat, dapatkan waktu epoch Unix saat ini dari mesin AWS IoT aturan dengan [stempel](#) waktu (). Fungsi ini menghasilkan waktu dalam milidetik, jadi Anda harus mengonversi nilainya menjadi waktu dalam hitungan detik dan mengimbangi dalam nanodetik. Untuk melakukannya, gunakan konversi berikut:

- Untuk Time in seconds (timeInSeconds), gunakan `${floor(timestamp() / 1E3)}` untuk mengubah waktu dari milidetik ke detik.
- Untuk Offset dalam nanos (offsetInNanos), gunakan `${(timestamp() % 1E3) * 1E6}` untuk menghitung offset nanodetik dari stempel waktu.

Mengonversi stempel waktu yang dalam format string

Jika sensor atau peralatan Anda melaporkan data waktu dalam format string (misalnya, `2020-03-03T14:57:14.699Z`), gunakan [time_to_epoch](#) (String, String). Fungsi ini memasukkan stempel waktu dan pola format sebagai parameter dan waktu keluaran dalam milidetik. Kemudian, Anda harus mengubah waktu ke waktu dalam hitungan detik dan mengimbangi dalam nanodetik. Untuk melakukannya, gunakan konversi berikut:

- Untuk Time in seconds (`timeInSeconds`), gunakan `${floor(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") / 1E3)}` untuk mengonversi string timestamp menjadi milidetik, lalu ke detik.
- Untuk Offset dalam nanos (`offsetInNanos`), gunakan `${(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") % 1E3) * 1E6}` untuk menghitung offset nanodetik dari string stempel waktu.

Note

`time_to_epoch` Fungsi ini mendukung string stempel waktu presisi hingga milidetik. Untuk mengonversi string dengan presisi mikrodetik atau nanodetik, konfigurasi AWS Lambda fungsi yang dipanggil aturan Anda untuk mengonversi stempel waktu menjadi nilai numerik. Untuk informasi selengkapnya, lihat [Mengonversi string stempel waktu presisi nanodetik](#).

Mengonversi string stempel waktu presisi nanodetik

Jika perangkat Anda mengirimkan informasi stempel waktu dalam format string dengan presisi nanodetik (misalnya, `2020-03-03T14:57:14.699728491Z`), gunakan prosedur berikut untuk mengonfigurasi tindakan aturan Anda. Anda dapat membuat AWS Lambda fungsi yang mengubah stempel waktu dari string menjadi Time in seconds (`timeInSeconds`) dan Offset in nanos (`offsetInNanos`). Kemudian, gunakan [aws_lambda \(FunctionARN, InputJson\) dalam parameter tindakan aturan Anda untuk menjalankan fungsi Lambda itu dan menggunakan output](#) dalam aturan Anda.

Note

Bagian ini berisi petunjuk lanjutan yang mengasumsikan bahwa Anda terbiasa dengan cara membuat sumber daya berikut:

- Fungsi Lambda. Untuk informasi selengkapnya, lihat [Membuat fungsi Lambda pertama Anda](#) di Panduan AWS Lambda Pengembang.
- AWS IoT aturan dengan tindakan AWS IoT SiteWise aturan. Untuk informasi selengkapnya, lihat [Menelan data untuk AWS IoT SiteWise menggunakan aturan AWS IoT Core](#).

Untuk membuat tindakan AWS IoT SiteWise aturan yang mem-parsing string stempel waktu

1. Buat fungsi Lambda dengan properti berikut:

- Nama fungsi — Gunakan nama fungsi deskriptif (misalnya, **ConvertNanosecondTimestampFromString**).
- Runtime — Gunakan runtime Python 3, seperti Python 3.11 (). `python3.11`
- Izin — Buat peran dengan izin Lambda dasar (). `AWS LambdaBasicExecutionRole`
- Layers - Tambahkan AWS SDKPandaslayer -Python311 untuk fungsi Lambda yang akan digunakan. `numpy`
- Kode fungsi — Gunakan kode fungsi berikut, yang mengkonsumsi argumen string bernama `timestamp` dan output `timeInSeconds` dan `offsetInNanos` nilai untuk stempel waktu itu.

```
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
}
```

[Fungsi Lambda ini memasukkan string stempel waktu dalam format ISO 8601 menggunakan `datetime64 from`. NumPy](#)

Note

Jika string stempel waktu Anda tidak dalam format ISO 8601, Anda dapat menerapkan solusi dengan `pandas` yang mendefinisikan format stempel waktu. Untuk informasi selengkapnya, lihat [pandas.to_datetime](#).

2. Saat Anda mengonfigurasi AWS IoT SiteWise tindakan untuk aturan Anda, gunakan templat substitusi berikut untuk Time in seconds (`timeInSeconds`) dan Offset in nanos (`offsetInNanos`). Template substitusi ini mengasumsikan bahwa payload pesan Anda berisi string stempel waktu. `timestamp` `aws_lambdaFungsi` ini menggunakan struktur JSON untuk parameter kedua, sehingga Anda dapat memodifikasi template substitusi di bawah ini jika diperlukan.

- Untuk Time in seconds (`timeInSeconds`), gunakan template substitusi berikut.

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

- Untuk Offset di nanos (`offsetInNanos`), gunakan template substitusi berikut.

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

Untuk setiap parameter, ganti `region` dan `account-id` dengan Wilayah dan ID AWS akun Anda. Jika Anda menggunakan nama yang berbeda untuk fungsi Lambda Anda, ubah juga.

3. Berikan AWS IoT izin untuk menjalankan fungsi Anda dengan izin. `lambda:InvokeFunction` Untuk informasi selengkapnya, lihat [aws_lambda \(FunctionARN, InputJson\)](#).

4. Uji aturan Anda (misalnya, gunakan klien pengujian AWS IoT MQTT) dan verifikasi bahwa AWS IoT SiteWise menerima data yang Anda kirim.

Jika aturan Anda tidak berfungsi seperti yang diharapkan, lihat [Memecahkan masalah tindakan aturan AWS IoT SiteWise](#).

Note

Solusi ini memanggil fungsi Lambda dua kali untuk setiap string stempel waktu. Anda dapat membuat aturan lain untuk mengurangi jumlah pemanggilan fungsi Lambda jika aturan Anda menangani beberapa titik data yang memiliki stempel waktu yang sama di setiap muatan. Untuk melakukannya, buat aturan dengan tindakan penerbitan ulang yang memanggil Lambda dan menerbitkan payload asli dengan string stempel waktu yang dikonversi ke `dan.timeInSeconds offsetInNanos`. Kemudian, buat aturan dengan tindakan AWS IoT SiteWise aturan untuk mengonsumsi payload yang dikonversi. Dengan pendekatan ini, Anda mengurangi berapa kali aturan memanggil Lambda tetapi meningkatkan jumlah tindakan aturan AWS IoT yang dijalankan. Pertimbangkan harga setiap layanan jika Anda menerapkan solusi ini untuk kasus penggunaan Anda.

Contoh konfigurasi aturan

Bagian ini berisi contoh konfigurasi aturan untuk membuat aturan dengan AWS IoT SiteWise tindakan.

Example Contoh tindakan aturan yang menggunakan alias properti sebagai topik pesan

Contoh berikut membuat aturan dengan AWS IoT SiteWise tindakan yang menggunakan topik (melalui [topik \(\)](#)) sebagai alias properti untuk mengidentifikasi properti aset. Gunakan contoh ini untuk menentukan satu aturan untuk menelan data tipe ganda ke semua turbin angin di semua ladang angin. Contoh ini mengharuskan Anda mendefinisikan alias properti pada semua properti aset turbin. Anda perlu mendefinisikan aturan kedua yang mirip dengan menelan data tipe bilangan bulat.

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

Contoh payload di `sitewise-rule-payload.json` berisi konten berikut.


```
{
  "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"
      }
    }
  ]
}
```

Dengan tindakan aturan ini, kirim pesan berikut ke alias properti turbin angin (misalnya, /company/windfarm/3/turbine/7/temperature) sebagai topik untuk menelan data.

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

Example Contoh aturan tindakan yang menggunakan `timestamp()` untuk menentukan waktu

Contoh berikut membuat aturan dengan AWS IoT SiteWise tindakan yang mengidentifikasi properti aset dengan IDs dan menggunakan [timestamp\(\)](#) untuk menentukan waktu saat ini.

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

Contoh payload di `sitewise-rule-payload.json` berisi konten berikut.

```
{
  "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"
    }
  ]
}
```

Dengan tindakan aturan ini, kirim pesan berikut `my/asset/property/topic` ke data untuk menelan.

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

Memecahkan masalah tindakan aturan

Untuk memecahkan masalah tindakan AWS IoT SiteWise aturan Anda AWS IoT Core, konfigurasi CloudWatch Log atau konfigurasi tindakan kesalahan penerbitan ulang untuk aturan Anda. Untuk informasi selengkapnya, lihat [Memecahkan masalah tindakan aturan AWS IoT SiteWise](#).

Kurangi biaya dengan Basic Ingest di AWS IoT SiteWise

AWS IoT Core [menyediakan fitur yang disebut Basic Ingest yang dapat Anda gunakan untuk mengirim data AWS IoT Core tanpa menimbulkan biaya pengiriman pesan.](#) AWS IoT Basic Ingest mengoptimalkan aliran data untuk beban kerja konsumsi data volume tinggi dengan menghapus broker pesan terbitkan/berlangganan dari jalur konsumsi. Anda dapat menggunakan Basic Ingest jika Anda tahu aturan mana pesan Anda harus diarahkan.

Untuk menggunakan Basic Ingest, Anda mengirim pesan langsung ke aturan tertentu menggunakan topik khusus. `$aws/rules/rule-name` Misalnya, untuk mengirim pesan ke aturan bernama `SiteWiseWindFarmRule`, Anda mengirim pesan ke topik `$aws/rules/SiteWiseWindFarmRule`.

Jika tindakan aturan Anda menggunakan templat substitusi yang berisi [topik \(Desimal\)](#), Anda dapat meneruskan topik asli di akhir topik khusus Basic Ingest, seperti. `$aws/rules/rule-name/original-topic` Misalnya, untuk menggunakan Basic Ingest dengan contoh alias properti ladang angin dari bagian sebelumnya, Anda dapat mengirim pesan ke topik berikut.

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

Note

Contoh di atas menyertakan garis miring kedua (//) karena AWS IoT menghapus awalan Basic Ingest (`$aws/rules/rule-name/`) dari topik yang terlihat oleh tindakan

aturan. Dalam contoh ini, aturan menerima topik/company/windfarm/3/turbine/7/temperature.

Untuk informasi selengkapnya, lihat [Mengurangi biaya pesan dengan konsumsi dasar](#) di Panduan AWS IoT Pengembang.

Menelan data ke dari AWS IoT SiteWiseAWS IoT Events

Dengan AWS IoT Events, Anda dapat membangun aplikasi pemantauan acara yang kompleks untuk armada IoT Anda di Cloud. AWS Gunakan SiteWise tindakan IoT AWS IoT Events untuk mengirim data ke properti aset AWS IoT SiteWise ketika suatu peristiwa terjadi.

Note

Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS IoT Events. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses AWS IoT Events konsol atau AWS IoT Events sumber daya. Untuk informasi lebih lanjut, lihat [AWS IoT Events akhir dukungan](#).

AWS IoT Events dirancang untuk merampingkan pengembangan aplikasi pemantauan acara untuk perangkat dan sistem IoT di dalam AWS Cloud. Menggunakan AWS IoT Events, Anda dapat:

- Mendeteksi dan merespons perubahan, anomali, atau kondisi spesifik di seluruh armada IoT Anda.
- Tingkatkan efisiensi operasional Anda dan aktifkan manajemen proaktif ekosistem IoT Anda.

Dengan mengintegrasikan dengan AWS IoT SiteWise melalui AWS IoT SiteWise tindakan, AWS IoT Events memperluas kemampuannya, memungkinkan Anda memperbarui properti aset secara otomatis sebagai respons terhadap peristiwa tertentu. AWS IoT SiteWise Interaksi ini dapat menyederhanakan konsumsi dan manajemen data. Ini juga dapat memberdayakan Anda dengan wawasan yang dapat ditindaklanjuti.

Untuk informasi selengkapnya, lihat topik berikut di Panduan AWS IoT Events Pengembang:

- [Apa itu AWS IoT Events?](#)
- [AWS IoT Events tindakan](#)

- [Tindakan IoT SiteWise](#)

Gunakan pengelola AWS IoT Greengrass aliran di AWS IoT SiteWise

AWS IoT Greengrass Stream Manager adalah fitur integrasi yang memfasilitasi transfer aliran data dari sumber lokal ke AWS Cloud. Ini bertindak sebagai lapisan perantara yang mengelola aliran data, memungkinkan perangkat yang beroperasi di tepi untuk mengumpulkan dan menyimpan data sebelum dikirim ke AWS IoT SiteWise, untuk analisis dan pemrosesan lebih lanjut.

Tambahkan tujuan data dengan mengonfigurasi sumber lokal di AWS IoT SiteWise konsol. Anda juga dapat menggunakan pengelola aliran dalam AWS IoT Greengrass solusi kustom Anda untuk menyerap data. AWS IoT SiteWise

Note

Untuk menyerap data dari sumber OPC UA, konfigurasi gateway AWS IoT SiteWise Edge yang berjalan. AWS IoT Greengrass Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise gateway Edge](#).

Untuk informasi selengkapnya tentang cara mengonfigurasi tujuan untuk data sumber lokal, lihat [Memahami tujuan AWS IoT SiteWise Edge](#).

Untuk informasi selengkapnya tentang cara menyerap data menggunakan pengelola aliran dalam AWS IoT Greengrass solusi khusus, lihat topik berikut di Panduan AWS IoT Greengrass Version 2 Pengembang:

- [Apa itu AWS IoT Greengrass?](#)
- [Mengelola aliran data pada inti AWS IoT Greengrass](#)
- [Mengekspor data ke properti AWS IoT SiteWise aset](#)

Gunakan AWS IoT SiteWise gateway Edge

AWS IoT SiteWise Edge memperluas kemampuan cloud ke lingkungan tepi industri, memungkinkan pemrosesan data lokal, analisis, dan pengambilan keputusan. SiteWise Edge terintegrasi dengan AWS IoT SiteWise dan AWS layanan lainnya untuk menyediakan solusi IoT industri yang komprehensif. Gateway berfungsi sebagai perantara antara peralatan industri Anda dan AWS IoT SiteWise.

SiteWise Edge gateway berjalan pada dua target penyebaran yang berbeda:

- AWS IoT Greengrass V2
- Siemens Industrial Edge

Anda dapat menggunakan gateway SiteWise Edge untuk mengumpulkan data di tepi dan mempublikasikannya ke cloud. Untuk gateway yang berjalan AWS IoT Greengrass, Anda juga dapat memproses data di edge menggunakan model aset dan aset.

Aplikasi AWS IoT SiteWise Edge Siemens Industrial Edge mendukung integrasi antara peralatan industri AWS IoT SiteWise sehingga Anda dapat mengumpulkan dan memproses data mesin mentah dan menjalankan analisis secara lokal sebelum mengirim data yang disempurnakan ke Cloud. AWS

Konsep kunci gateway SiteWise Edge

SiteWise Edge memiliki beberapa fitur yang berguna untuk komputasi tepi di lingkungan industri.

Pengumpulan dan pemrosesan data lokal

Mendukung pengumpulan data dari aset industri menggunakan protokol seperti OPC-UA dan MQTT. Gateway berjalan di perangkat AWS IoT Greengrass Core atau Siemens Industrial Edge

Operasi offline

Terus mengumpulkan dan memproses data selama pemadaman internet, menyinkronkan dengan cloud saat konektivitas dipulihkan.

Komputasi tepi dengan AWS IoT Greengrass komponen

Menggunakan SiteWise penerbit IoT untuk meneruskan data ke cloud dan AWS IoT SiteWise prosesor untuk transformasi dan perhitungan lokal. Baik penerbit dan prosesor adalah AWS

IoT Greengrass V2 komponen. Untuk informasi selengkapnya tentang AWS IoT Greengrass komponen, lihat [komponen AWS yang disediakan](#).

Integrasi dengan AWS IoT SiteWise untuk memperluas fitur cloud

Bekerja dengan fitur AWS IoT SiteWise cloud, memperluas model aset, analitik, dan dasbor ke tepi.

Untuk gateway dengan paket pemrosesan data yang diaktifkan, Anda dapat menggunakannya AWS OpsHub AWS IoT SiteWise untuk mengelola gateway Edge secara terpusat. SiteWise AWS OpsHub menyediakan kemampuan manajemen dan pemantauan jarak jauh. Untuk informasi selengkapnya, lihat [Kelola gateway SiteWise Edge menggunakan for AWS OpsHubAWS IoT SiteWise](#).

Integrasi sumber data mitra

Hubungkan sumber data mitra ke gateway Anda dan terima data dari mitra di gateway SiteWise Edge dan AWS cloud Anda. Untuk informasi selengkapnya, lihat [Sumber data mitra di gateway SiteWise Edge](#).

Visualisasi lokal di tepi

Menyediakan dasbor khusus untuk wawasan waktu nyata di tepi.

Pantau data secara lokal di fasilitas Anda menggunakan portal SiteWise Monitor di perangkat lokal Anda. Untuk informasi selengkapnya, lihat [Mengaktifkan AWS IoT SiteWise portal Anda di tepi](#).

Manfaat mengimplementasikan SiteWise Edge

SiteWise Edge menawarkan banyak keuntungan yang secara signifikan dapat meningkatkan operasi industri dan proses pengambilan keputusan.

- Wawasan operasional real-time tanpa penundaan pemrosesan cloud
- Kontinuitas operasional di lingkungan yang terputus
- Mengurangi biaya bandwidth dan penyimpanan melalui pra-pemrosesan tepi
- Peningkatan keandalan dengan kemampuan untuk membuat keputusan lokal berdasarkan data

Self-host gateway AWS IoT SiteWise Edge dengan AWS IoT Greengrass V2

Siapkan AWS IoT SiteWise Edge untuk mengumpulkan, memproses, dan memvisualisasikan data dari peralatan industri secara lokal sebelum mengirimkannya ke cloud. Self-host menggunakan AWS IoT Greengrass Version 2.

Gerbang AWS IoT SiteWise Edge bertindak sebagai perantara antara peralatan industri Anda dan AWS IoT SiteWise. Berjalan AWS IoT Greengrass Version 2, gateway SiteWise Edge mendukung pengumpulan dan pemrosesan data di tempat. Pantau data secara lokal di fasilitas Anda melalui portal SiteWise Monitor di perangkat lokal Anda dengan paket pemrosesan data diaktifkan dan AWS OpsHub diinstal.

Ada dua jenis gateway yang dihosting sendiri:

Berkemampuan MQTT, gateway V3

Arsitektur gateway V3 berkemampuan MQTT menyediakan kemampuan konsumsi data yang ditingkatkan. Ini menggunakan protokol MQTT untuk komunikasi data yang efisien dan menawarkan tujuan data yang dapat dikonfigurasi. Ini termasuk opsi untuk penyerapan data buffer menggunakan Amazon S3, serta konsumsi data waktu nyata. Anda dapat menerapkan filter jalur untuk berlangganan topik MQTT tertentu, memungkinkan pengumpulan data yang ditargetkan. Perhatikan bahwa gateway V3 berkemampuan MQTT tidak mendukung fitur Paket Pemrosesan Data. Untuk informasi selengkapnya, lihat [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#).

Aliran klasik, gateway V2

Aliran Klasik, gateway V2 mewakili arsitektur gateway AWS IoT SiteWise Edge tradisional. Ini sangat cocok untuk penerapan SiteWise Edge yang ada dan pengguna yang terbiasa dengan alur kerja yang ditetapkan. Sementara aliran Klasik, gateway V2 mendukung paket pemrosesan data, perhatikan bahwa data yang dihasilkan oleh paket pemrosesan data tidak dapat dicerna melalui Amazon S3. Gunakan aliran Klasik, gateway V2 jika Anda perlu mempertahankan kompatibilitas dengan penerapan yang ada atau jika Anda memerlukan fungsionalitas paket pemrosesan data. Untuk informasi selengkapnya, lihat [Aliran klasik, gateway V2 untuk Edge AWS IoT SiteWise](#).

Topik

- [AWS IoT SiteWise Persyaratan gateway yang dihosting sendiri di Edge](#)

- [Buat gateway SiteWise Edge yang dihosting sendiri](#)
- [Instal perangkat lunak gateway AWS IoT SiteWise Edge di perangkat lokal Anda](#)
- [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#)
- [Aliran klasik, gateway V2 untuk Edge AWS IoT SiteWise](#)
- [Tambahkan sumber data ke gateway AWS IoT SiteWise Edge Anda](#)
- [AWS IoT Greengrass komponen untuk AWS IoT SiteWise Edge](#)
- [Filter aset di gateway SiteWise Edge](#)
- [Konfigurasi dukungan proxy dan kelola toko kepercayaan untuk AWS IoT SiteWise Edge](#)
- [Gunakan AWS IoT SiteWise APIs di tepi](#)

AWS IoT SiteWise Persyaratan gateway yang dihosting sendiri di Edge

AWS IoT SiteWise Edge gateway berjalan AWS IoT Greengrass V2 sebagai satu set AWS IoT Greengrass komponen yang mendukung pengumpulan data, pemrosesan, dan penerbitan di tempat. Untuk mengonfigurasi gateway SiteWise Edge yang berjalan AWS IoT Greengrass V2, buat gateway di AWS Cloud dan jalankan perangkat lunak gateway SiteWise Edge untuk menyiapkan perangkat lokal Anda. Saat Anda menggunakan AWS Management Console untuk membuat gateway SiteWise Edge, skrip instalasi disediakan. Jalankan skrip ini di perangkat gateway target Anda untuk menyiapkan perangkat lunak dan dependensi yang diperlukan.

Persyaratan perangkat lokal

Perangkat lokal harus memenuhi persyaratan berikut untuk menginstal dan menjalankan perangkat lunak gateway SiteWise Edge.

- Mendukung perangkat lunak AWS IoT Greengrass V2 Core versi [v2.3.0](#) atau yang lebih baru. Untuk informasi selengkapnya, lihat [Persyaratan](#) dalam Panduan AWS IoT Greengrass Version 2 Pengembang.

- Salah satu platform yang didukung berikut:

- OS: Ubuntu 20.04 atau yang lebih baru

Arsitektur: x86_64 (AMD64) atau ARMv8 (Aarch64)

- OS: Perusahaan Red Hat Linux (RHEL) 8

Arsitektur: x86_64 (AMD64) atau ARMv8 (Aarch64)

- OS: Amazon Linux 2

Arsitektur: x86_64 (AMD64) atau ARMv8 (Aarch64)

- OS: Debian 11

Arsitektur: x86_64 (AMD64) atau ARMv8 (Aarch64)

- OS: Windows Server 2019 dan yang lebih baru

Arsitektur: x86_64 () AMD64

Note

Platform ARM mendukung gateway SiteWise Edge hanya dengan Paket Pengumpulan Data. Paket pemrosesan data tidak didukung.

- Minimal 4 GB RAM.
- Ruang disk minimal 10 GB tersedia untuk perangkat lunak gateway SiteWise Edge.
- Konfigurasi perangkat lokal Anda untuk memastikan bahwa port yang tepat dapat diakses. Untuk daftar lengkap titik akhir layanan keluar yang diperlukan, lihat Titik akhir layanan yang [diperlukan untuk AWS IoT SiteWise gateway Edge](#).
- Java Runtime Environment (JRE) versi 11 atau lebih tinggi. Java harus tersedia pada variabel PATH lingkungan pada perangkat. Untuk menggunakan Java untuk mengembangkan komponen kustom, Anda harus instal Java Development Kit (JDK). [Kami menyarankan Anda menggunakan Amazon Corretto atau OpenJDK](#).

Bucket Amazon S3 untuk mengizinkan daftar perangkat lokal

Konfigurasi perangkat lokal Anda untuk menyediakan akses firewall ke bucket Amazon S3 berikut. Konfigurasi akses berdasarkan wilayah masing-masing untuk perangkat Anda.

Wilayah	Titik akhir
Asia Pasifik (Tokyo)	<code>https://iot-sitewise-gateway-ap-northeast-1-785558802005.s3.ap-northeast-1.amazonaws.com</code>
Asia Pacific (Seoul)	<code>https://iot-sitewise-gateway-ap-northeast-2-310055672453.s3.ap-northeast-2.amazonaws.com</code>

Wilayah	Titik akhir
Asia Pacific (Mumbai)	https://iot-sitewise-gateway-ap-south-1-677656657204.s3.ap-south-1.amazonaws.com
Asia Pasifik (Singapura)	https://iot-sitewise-gateway-ap-southeast-1-475191558554.s3.ap-southeast-1.amazonaws.com
Asia Pacific (Sydney)	https://iot-sitewise-gateway-ap-southeast-2-396319432685.s3.ap-southeast-2.amazonaws.com
Kanada (Pusat)	https://iot-sitewise-gateway-ca-central-1-842060018567.s3.ca-central-1.amazonaws.com
Tiongkok (Beijing)	https://iot-sitewise-gateway-cn-north-1-237124890262.s3.cn-north-1.amazonaws.com.cn
Eropa (Frankfurt)	https://iot-sitewise-gateway-eu-central-1-748875242063.s3.eu-central-1.amazonaws.com
Eropa (Irlandia)	https://iot-sitewise-gateway-eu-west-1-383414315062.s3.eu-west-1.amazonaws.com
AS Timur (Virginia Utara)	https://iot-sitewise-gateway-us-east-1-223558168232.s3.us-east-1.amazonaws.com dan https://iot-sitewise-gateway-us-east-1-223558168232.s3.amazonaws.com/
AS Timur (Ohio)	https://iot-sitewise-gateway-us-east-2-005072661813.s3.us-east-2.amazonaws.com
AWS GovCloud (AS-Barat)	https://iot-sitewise-gateway-us-gov-west-1-599984565679.s3.us-gov-west-1.amazonaws.com/
AS Barat (Oregon)	https://iot-sitewise-gateway-us-west-2-502577205460.s3.us-west-2.amazonaws.com

Persyaratan paket pemrosesan data

- Jika Anda berencana untuk menggunakan paket pemrosesan data di tepi AWS IoT SiteWise, perangkat lokal Anda juga harus memenuhi persyaratan berikut:
 - Memiliki prosesor quad-core x86 64 bit.
 - Memiliki setidaknya 16 GB RAM.
 - Memiliki setidaknya 32 GB untuk RAM jika menggunakan Microsoft Windows.
 - Memiliki setidaknya 256 GB ruang disk kosong.
 - Perangkat lokal harus mengizinkan lalu lintas masuk jaringan pada port 443.
 - Port berikut dicadangkan untuk digunakan oleh AWS IoT SiteWise: 80, 443, 3001, 4569, 4572, 8000, 8081, 8082, 8084, 8085, 8445, 8086, 9000, 9500, 11080, dan 50010. Menggunakan port cadangan untuk lalu lintas dapat mengakibatkan koneksi terputus.

Note

Komponen AWS IoT Greengrass V2 Stream manager memiliki persyaratannya sendiri. Untuk informasi selengkapnya, lihat [Konfigurasi](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

- Ruang disk minimum dan persyaratan kapasitas komputasi bergantung pada berbagai faktor yang unik untuk implementasi dan kasus penggunaan Anda.
- Ruang disk yang diperlukan untuk caching data untuk konektivitas internet intermiten tergantung pada faktor-faktor berikut:
 - Jumlah aliran data yang diunggah
 - Titik data per aliran data per detik
 - Ukuran setiap titik data
 - Kecepatan komunikasi
 - Waktu henti jaringan yang diharapkan
- Kapasitas komputasi yang diperlukan untuk melakukan polling dan mengunggah data tergantung pada faktor-faktor berikut:
 - Jumlah aliran data yang diunggah
 - Titik data per aliran data per detik

Konfigurasi izin untuk menggunakan gateway SiteWise Edge

Anda harus memiliki izin berikut untuk menggunakan gateway SiteWise Edge:

Note

Jika Anda menggunakan AWS IoT SiteWise konsol untuk membuat gateway SiteWise Edge Anda, izin ini ditambahkan untuk Anda.

- Peran IAM untuk gateway SiteWise Edge Anda harus memungkinkan Anda menggunakan gateway SiteWise Edge pada AWS IoT Greengrass V2 perangkat untuk memproses data model aset dan data aset.

Peran tersebut memungkinkan layanan berikut untuk mengambil `peran:credentials.iot.amazonaws.com`.

Detail izin

Peran harus memiliki izin berikut:

- `iotsitewise`— Memungkinkan prinsipal untuk mengambil data model aset dan data aset di tepi.
- `iot`— Memungkinkan AWS IoT Greengrass V2 perangkat Anda untuk berinteraksi dengan AWS IoT.
- `logs`— Memungkinkan AWS IoT Greengrass V2 perangkat Anda mengirim log ke Amazon CloudWatch Logs.
- `s3`— Memungkinkan AWS IoT Greengrass V2 perangkat Anda mengunduh artefak komponen khusus dari Amazon S3.

```
{
  "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": "*"
  }
]
```

Buat gateway SiteWise Edge yang dihosting sendiri

Gunakan AWS IoT SiteWise konsol atau AWS CLI untuk membuat gateway SiteWise Edge yang dihosting sendiri. Prosedur ini merinci cara membuat gateway SiteWise Edge yang dihosting sendiri yang akan Anda instal di perangkat keras Anda sendiri. Untuk informasi tentang membuat gateway SiteWise Edge yang berjalan di Siemens Industrial Edge, lihat. [Host gateway SiteWise Edge di Siemens Industrial Edge](#)

Buat gateway SiteWise Edge

Console

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih Buat gateway.
4. Untuk Pilih target penerapan, pilih gateway yang dihosting sendiri.

5. Pilih salah satu berkemampuan MQTT, gateway V3 atau aliran Klasik, gateway V2. Untuk informasi lebih lanjut tentang setiap opsi, lihat [Self-host gateway AWS IoT SiteWise Edge dengan AWS IoT Greengrass V2](#). Gateway V3 berkemampuan MQTT direkomendasikan untuk fitur-fiturnya yang siap di masa depan.
6. Di bagian konfigurasi Gateway, masukkan nama untuk gateway SiteWise Edge Anda atau gunakan nama yang dihasilkan oleh AWS IoT SiteWise.
7. Di bawah Greengrass device OS, pilih sistem operasi perangkat tempat Anda akan menginstal gateway Edge ini. SiteWise

Note

Paket pemrosesan data hanya tersedia di platform x86. Ini hanya tersedia di aliran Klasik, gateway V2

8. (Opsional) Untuk memproses dan mengatur data di tepi, di bawah kemampuan Edge, pilih Paket Pemrosesan Data.

Note

Untuk memberikan akses kepada grup pengguna di direktori perusahaan Anda ke gateway SiteWise Edge ini, lihat [Siapkan kemampuan tepi di SiteWise Edge](#)

9. (Opsional) Di bawah konfigurasi lanjutan, lakukan hal berikut:
 - Untuk perangkat inti Greengrass, pilih salah satu opsi berikut:
 - Pengaturan default - AWS secara otomatis menggunakan pengaturan default untuk membuat perangkat inti Greengrass di. AWS IoT Greengrass V2
 1. Masukkan nama untuk perangkat inti Greengrass atau gunakan nama yang dihasilkan oleh. AWS IoT SiteWise
 - Pengaturan lanjutan - Pilih opsi ini jika Anda ingin menggunakan perangkat inti Greengrass yang ada atau membuatnya secara manual.
 1. Pilih perangkat inti Greengrass atau pilih Buat perangkat inti Greengrass untuk membuatnya di konsol. AWS IoT Greengrass V2 Untuk informasi selengkapnya, lihat [Menyiapkan perangkat AWS IoT Greengrass V2 inti](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

10. Pilih Buat gateway.

11. Dalam kotak dialog Generate SiteWise Edge gateway installer, pilih Hasilkan dan unduh. AWS IoT SiteWise secara otomatis menghasilkan penginstal yang dapat Anda gunakan untuk mengonfigurasi perangkat lokal Anda.

 Important

Anda tidak dapat membuat ulang file ini. Pastikan Anda menyimpan file installer di lokasi yang aman karena Anda akan menggunakan file tersebut nanti.

AWS CLI

Untuk membuat gateway yang dihosting sendiri dengan menggunakan AWS CLI, berikan nama untuk gateway, tentukan platform, dan versi gateway. Ada banyak opsi lain yang dapat Anda tentukan saat membuat gateway. Untuk informasi selengkapnya, lihat [create-gateway](#) di Command Reference untuk AWS CLI AWS IoT SiteWise

Untuk menggunakan contoh ini, ganti placeholder input pengguna dengan informasi Anda sendiri.

```
aws iotsitewise create-gateway \  
  --gateway-name your-gateway-name \  
  --gateway-platform greengrassV2={coreDeviceThingName=your-core-device-thing-name, coreDeviceOperatingSystem=LINUX_AMD64} \  
  --gateway-version 3 \  
  [--cli-input-json your-configuration]
```

- `gateway-name`— Nama unik untuk gateway.
- `gateway-platform`— Menentukan konfigurasi platform gateway. Untuk gateway yang dihosting sendiri, masukkan `greengrassV2`. Untuk informasi selengkapnya, lihat [Ops](#) di bagian `create-gateway` dari Referensi AWS CLI Perintah untuk AWS IoT SiteWise
- `coreDeviceThingName`— Nama hal AWS IoT untuk perangkat AWS IoT Greengrass V2 inti Anda.
- `coreDeviceOperatingSystem`— Sistem operasi perangkat inti di AWS IoT Greengrass V2. Menentukan sistem operasi diperlukan untuk `gateway-version 3` dan tidak berlaku untuk `gateway-version 2`. Pilihannya meliputi: `LINUX_AARCH64`, `LINUX_AMD64`, dan `WINDOWS_AMD64`
- `gateway-version`— Versi gateway.

- Untuk membuat gateway V3 berkemampuan MQTT, gunakan 3 untuk versi gateway.
- Untuk membuat aliran Klasik, gateway V2, gunakan 2 untuk versi gateway.
- `cli-input-json`- File JSON yang berisi parameter permintaan.

Sekarang Anda telah membuat gateway SiteWise Edge, [Instal perangkat lunak gateway AWS IoT SiteWise Edge di perangkat lokal Anda](#).

Instal perangkat lunak gateway AWS IoT SiteWise Edge di perangkat lokal Anda

Setelah Anda membuat gateway AWS IoT SiteWise Edge, instal perangkat lunak gateway SiteWise Edge di perangkat lokal Anda. SiteWise Perangkat lunak gateway Edge dapat diinstal pada perangkat lokal yang memiliki sistem operasi Linux atau Microsoft Windows server yang diinstal.

Important

Pastikan perangkat lokal Anda terhubung ke internet.

Linux

Prosedur berikut menggunakan SSH untuk terhubung ke perangkat lokal Anda. Atau, Anda dapat menggunakan USB flash drive atau alat lain untuk mentransfer file installer ke perangkat lokal Anda. Jika Anda tidak ingin menggunakan SSH, lewati ke Langkah 2: Instal perangkat lunak gateway SiteWise Edge di bawah ini.

Prasyarat SSH

Sebelum Anda terhubung ke perangkat Anda menggunakan SSH, lengkapi prasyarat berikut.

- Linux dan macOS - Unduh dan instal OpenSSH. Untuk informasi selengkapnya, lihat <https://www.openssh.com>.

Langkah 1: Salin penginstal ke perangkat gateway SiteWise Edge Anda

Petunjuk berikut menjelaskan cara menghubungkan ke perangkat lokal Anda menggunakan klien SSH.

1. Untuk terhubung ke perangkat Anda, jalankan perintah berikut di jendela terminal di komputer Anda, ganti *username* dan *IP* dengan nama pengguna yang memiliki hak istimewa dan alamat IP yang tinggi.

```
ssh username@IP
```

2. Untuk mentransfer file penginstal yang AWS IoT SiteWise dihasilkan ke perangkat gateway SiteWise Edge Anda, jalankan perintah berikut.

 Note

- Ganti *path-to-saved-installer* dengan path di komputer Anda yang Anda gunakan untuk menyimpan file installer dan nama file installer.
- Ganti *IP-address* dengan alamat IP perangkat lokal Anda.
- Ganti *directory-to-receive-installer* dengan jalur di perangkat lokal yang Anda gunakan untuk menerima file penginstal.

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

Langkah 2: Instal perangkat lunak gateway SiteWise Edge

Dalam prosedur berikut, jalankan perintah di jendela terminal pada perangkat gateway SiteWise Edge Anda.

1. Berikan file penginstal izin eksekusi.

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

2. Jalankan pemasang.

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

Windows Server

Prasyarat

Anda harus memiliki prasyarat berikut untuk menginstal perangkat lunak gateway Edge: SiteWise

- Microsoft WindowsServer 2019 atau yang lebih baru diinstal
- Hak istimewa administrator
- PowerShell versi 5.1 atau yang lebih baru diinstal
- SiteWise Edge gateway installer diunduh ke Windows Server di mana ia akan disediakan

Langkah 1: Jalankan PowerShell sebagai administrator

1. Di server Windows tempat Anda ingin menginstal gateway SiteWise Edge, masuk sebagai administrator.
2. Masukkan PowerShell di bilah pencarian Windows.
3. Dalam hasil pencarian, buka menu konteks (klik kanan) pada PowerShell aplikasi Windows. Pilih Jalankan sebagai Administrator.

Langkah 2: Instal perangkat lunak gateway SiteWise Edge

Jalankan perintah berikut di jendela terminal pada perangkat SiteWise Edge Gateway Anda.

1. Buka blokir penginstal gateway SiteWise Edge.

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

2. Jalankan Installer.

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

 Note

Jika eksekusi skrip dinonaktifkan pada sistem, ubah kebijakan eksekusi skrip menjadi `RemoteSigned`.

```
Set-ExecutionPolicy RemoteSigned
```

Langkah selanjutnya tergantung pada jenis gateway yang dihosting sendiri yang Anda butuhkan. Lanjutkan ke [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#) atau [Aliran klasik, gateway V2 untuk Edge AWS IoT SiteWise](#).

Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise

AWS IoT SiteWise dapat menggunakan gateway V3 berkemampuan MQTT, yang mewakili kemajuan signifikan dalam arsitektur gateway Edge. SiteWise Tipe gateway ini memanfaatkan protokol MQTT (Message Queuing Telemetry Transport) untuk komunikasi data, memberikan peningkatan fleksibilitas dan efisiensi dalam penerapan IoT industri.

Gateway V3 berkemampuan MQTT menggunakan MQTT untuk transfer data, memungkinkan protokol jaringan berlangganan publikasi yang ringan yang secara efisien mengangkut pesan antar perangkat dan cloud. Anda dapat mengatur berbagai tujuan data, termasuk konsumsi data real-time langsung ke dalam AWS IoT SiteWise dan penyerapan data buffer menggunakan Amazon S3. Untuk mengaktifkan pengumpulan data yang tepat, Anda dapat menerapkan filter jalur untuk berlangganan topik MQTT tertentu.

Gateway V3 berkemampuan MQTT dilengkapi dengan tujuan real-time yang telah dikonfigurasi sebelumnya dengan filter yang disetel ke “#” (semua topik), yang dapat Anda sesuaikan atau hapus sesuai kebutuhan. Untuk merampingkan manajemen data, hanya satu tujuan real-time yang dapat ada di setiap gateway.

Arsitektur berkemampuan MQTT berbeda secara signifikan dari aliran Klasik, gateway V2. Sementara V2 menggunakan pendekatan berbasis streaming, V3 menggunakan MQTT, menawarkan lebih banyak tujuan data yang dapat dikonfigurasi dan opsi pemfilteran. Namun, perhatikan bahwa V3 tidak mendukung paket pemrosesan data, yang tersedia di V2.

Gateway V3 berkemampuan MQTT menawarkan beberapa keuntungan:

- Peningkatan skalabilitas, karena sifat ringan MQTT, memungkinkan penanganan yang lebih baik dari berbagai perangkat dan transmisi data frekuensi tinggi.
- Kontrol data yang ditingkatkan melalui filter jalur, memungkinkan manajemen pengumpulan data secara terperinci dan mengurangi transfer dan pemrosesan data yang tidak perlu.
- Penanganan data yang fleksibel, memungkinkan konfigurasi antara pemrosesan real-time dan penyimpanan buffer berdasarkan kebutuhan spesifik.
- Keselarasan dengan standar komunikasi IoT modern, menetapkan panggung untuk peningkatan dan integrasi di masa depan.

Pertimbangkan untuk mengadopsi gateway V3 berkemampuan MQTT untuk penerapan baru, terutama ketika Anda memerlukan opsi penyerapan data yang fleksibel dan kontrol yang tepat atas pengumpulan data.

 Note

Untuk penerapan atau skenario yang ada yang memerlukan paket pemrosesan data, aliran Klasik, gateway V2 tetap menjadi opsi yang layak.

Dengan menawarkan kedua jenis gateway, AWS IoT SiteWise pastikan Anda dapat memilih solusi yang paling sesuai dengan kebutuhan IoT industri spesifik Anda, baik Anda memprioritaskan kemampuan MQTT tingkat lanjut atau kompatibilitas dengan sistem yang ada.

Tujuan dan filter jalur

Lihat topik berikut untuk mempelajari lebih lanjut tentang tujuan dan filter jalur di gateway berkemampuan MQTT:

- [Memahami tujuan AWS IoT SiteWise Edge](#)
- [Tambahkan tujuan real-time AWS IoT SiteWise Edge](#)
- [Tambahkan tujuan AWS IoT SiteWise buffer menggunakan Amazon S3](#)
- [Memahami filter jalur untuk tujuan AWS IoT SiteWise Edge](#)
- [Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge](#)
- [Kelola tujuan AWS IoT SiteWise Edge](#)

Connect aplikasi eksternal ke broker EMQX

Panduan ini menjelaskan cara menghubungkan aplikasi eksternal ke gateway AWS IoT SiteWise Edge Anda melalui broker EMQX di gateway V3 yang diaktifkan MQTT yang Anda gunakan. Aplikasi eksternal mungkin termasuk alat pemantauan khusus, perangkat lunak visualisasi pihak ketiga, atau sistem lama yang perlu berinteraksi dengan data industri Anda di edge.

Kami akan membahas langkah-langkah konfigurasi untuk Linux dan Microsoft Windows lingkungan, termasuk konfigurasi penerapan EMQX, pengaturan TLS untuk koneksi aman, dan aturan otorisasi untuk mengontrol akses ke topik tertentu.

 Note

EMQX bukan vendor atau pemasok untuk AWS IoT SiteWise Edge.

 Important

Untuk mengamankan koneksi ke gateway Anda, kami sangat menyarankan menggunakan otentikasi berbasis sertifikat melalui fitur otentikasi perangkat klien. AWS IoT Greengrass Metode ini memberikan keamanan yang kuat melalui otentikasi TLS (mTLS) timbal balik. Untuk informasi selengkapnya, lihat [Connect perangkat klien ke perangkat inti](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Jika Anda tidak dapat menggunakan otentikasi berbasis sertifikat, ikuti panduan ini untuk mengatur otentikasi menggunakan nama pengguna dan kata sandi.

Prasyarat

- Gateway V3 berkemampuan SiteWise Edge MQTT yang telah digunakan dan sedang online
- Akses ke host gateway
- Akses ke AWS IoT SiteWise dan AWS IoT Greengrass konsol

Topik

- [Format payload pesan untuk broker EMQX di Edge AWS IoT SiteWise](#)
- [Konfigurasi broker EMQX](#)
- [Hubungkan aplikasi ke broker EMQX di Edge AWS IoT SiteWise](#)
- [Siapkan aturan otorisasi untuk AWS IoT SiteWise Edge di EMQX](#)

Format payload pesan untuk broker EMQX di Edge AWS IoT SiteWise

Agar komponen SiteWise penerbit IoT mengkonsumsi data dari aplikasi eksternal Anda dan mempublikasikannya ke AWS IoT SiteWise cloud, muatan yang dikirim ke broker harus memenuhi persyaratan khusus.

Memahami format payload adalah kunci keberhasilan komunikasi MQTT dengan Edge. AWS IoT SiteWise Sementara proses pengaturan koneksi tercakup dalam bagian selanjutnya, kami

menyajikan persyaratan payload terlebih dahulu untuk membantu Anda merencanakan implementasi Anda.

Persyaratan topik MQTT

Tidak ada batasan pada struktur topik MQTT, termasuk jumlah level atau karakter yang digunakan. Namun, kami menyarankan agar topik tersebut cocok dengan `propertyAlias` bidang di payload.

Example Contoh alias properti

Jika topik MQTT adalah `site1/line1/compressor1/temperature`, pastikan kecocokannya. `propertyAlias`

```
{
  "assetId": "compressor_asset_01",
  "propertyAlias": "site1/line1/compressor1/temperature",
  "propertyId": "temperature_sensor_01",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "offsetInNanos": 0,
        "timeInSeconds": 1683000000
      },
      "value": {
        "doubleValue": 23.5
      }
    }
  ]
}
```

Struktur muatan JSON

[Payload pesan MQTT ditulis dalam JSON dan mengikuti format PutAssetPropertyValueEntry pesan yang ditentukan dalam Referensi API.AWS IoT SiteWise](#)

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
```

```
    "quality": "string",
    "timestamp": {
      "offsetInNanos": number,
      "timeInSeconds": number
    },
    "value": {
      "booleanValue": boolean,
      "doubleValue": number,
      "integerValue": number,
      "stringValue": "string"
    }
  }
]
```

Note

Agar pesan dianggap valid, hanya satu dari kondisi berikut yang bisa benar:

- yang `propertyAlias` ditetapkan, atau
- Keduanya `assetId` dan `propertyId` diatur

`PutAssetPropertyValueEntry` memiliki `entryId` bidang yang tidak diperlukan dalam konteks ini.

Konfigurasi broker EMQX

Bagian ini mencakup cara menambahkan nama pengguna dan kata sandi. Ini juga mencakup cara membuat koneksi TLS dari sumber eksternal menggunakan nama pengguna dan kata sandi yang ditambahkan. Anda dapat mengkonfigurasi broker EMQX menggunakan Linux atau Microsoft Windows

Note

Untuk mengonfigurasi broker, Anda memerlukan perangkat inti yang diatur dengan konfigurasi EMQX default di gateway V3 berkemampuan MQTT Anda.

⚠ Important

Setelah menyelesaikan prosedur ini, kami sangat menyarankan untuk mengonfigurasi aturan otorisasi. Untuk informasi selengkapnya, lihat [Siapkan aturan otorisasi untuk AWS IoT SiteWise Edge di EMQX](#). Aturan otorisasi untuk pengguna tambahan meningkatkan keamanan.

Perbarui konfigurasi penerapan EMQX untuk otentikasi

Untuk memperbarui konfigurasi penerapan EMQX untuk otentikasi

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway untuk dikonfigurasi.
4. Di bagian konfigurasi gateway Edge, salin nilai perangkat inti Greengrass Anda. Simpan untuk digunakan nanti.
5. Buka [konsol AWS IoT](#).
6. Di navigasi kiri, di bawah bagian Kelola, pilih Perangkat Greengrass, lalu Deployment.
7. Temukan nilai perangkat inti yang Anda simpan sebelumnya dan pilih tautan itu untuk membuka penerapan.
8. Pilih tombol dropdown Actions, lalu Revise.
9. Baca pesan yang muncul lalu pilih Revise deployment. Halaman Tentukan target muncul.
10. Pilih Berikutnya hingga Anda mencapai langkah Konfigurasi komponen.
11. Pilih tombol `aws.greengrass.clientdevices.mqtt.EMQX radio`.
12. Pilih tombol Configure component. Halaman konfigurasi muncul untuk komponen.
13. Di bawah Pembaruan konfigurasi, pilih Atur ulang ke konfigurasi default untuk versi komponen: `2.*.*`.
14. Masukkan konfigurasi berikut di bagian Configuration to merge berdasarkan OS Anda.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    }
  }
}
```

```

    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    },
    "authentication": {
      "enable": true,
      "backend": "built_in_database",
      "mechanism": "password_based",
      "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
      },
      "user_id_type": "username"
    },
    "dashboard": {
      "listeners": {
        "http": {
          "bind": 18083
        }
      }
    }
  },
  "authMode": "bypass",
  "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883\n-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e\nEMQX_NODE__NAME=emqx@local",
  "requiresPrivilege": "true"

```

```
}
```

Windows

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    },
    "authentication": {
      "enable": true,
      "backend": "built_in_database",
      "mechanism": "password_based",
      "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
      },
      "user_id_type": "username"
    },
    "dashboard": {
      "listeners": {
        "http": {
          "bind": 18083
        }
      }
    }
  }
}
```

```
    }  
  },  
  "authMode": "bypass",  
  "requiresPrivilege": "true"  
}
```

`dockerOptionsBidang` ini hanya untuk gateway Linux.

15. Pilih Konfirmasi.
16. Pilih Berikutnya sampai Anda mencapai langkah Review.
17. Pilih Deploy.
18. Setelah penerapan berhasil, lanjutkan ke langkah berikutnya.

Aktifkan otentikasi nama pengguna dan kata sandi

Bagian ini menunjukkan cara menambahkan nama pengguna dan kata sandi melalui GUI dasbor EMQX.

Note

Instruksi terkait EMQX yang disediakan hanya untuk referensi. Karena dokumentasi dan fitur EMQX dapat berubah seiring waktu, dan kami tidak memelihara dokumentasinya, kami sarankan untuk berkonsultasi dengan [dokumentasi resmi EMQX untuk informasi](#) terbaru.

EMQX Dashboard

Untuk mengaktifkan otentikasi nama pengguna dan kata sandi melalui dasbor EMQX

1. Pastikan Anda berada di dalam host gateway.
2. Buka jendela browser dan kunjungi <http://localhost:18083/>.
3. Masukkan nama pengguna default **admin** dan kata sandi default **public**. Untuk informasi selengkapnya, lihat [Dasbor EMQX di Dokumen](#) EMQX.
4. Setelah login, Anda diminta untuk mengubah kata sandi Anda. Perbarui kata sandi Anda untuk melanjutkan ke Dasbor EMQX.
5. Di navigasi kiri, pilih ikon perisai, lalu Otentikasi.
6. Di baris Built-in Database, pilih tombol Users.

7. Pilih tombol ikon tanda tambah untuk menambahkan pengguna. Layar Tambah muncul.
8. Masukkan nama pengguna dan kata sandi untuk pengguna aplikasi eksternal.
9. Pilih Simpan. Nama pengguna yang Anda pilih muncul di tabel halaman Otentikasi.

 Note

Aturan otorisasi yang ada atau default berlaku untuk pengguna baru. Disarankan untuk meninjau dan menyesuaikannya dengan kebutuhan aplikasi eksternal Anda.

EMQX Management with Linux

Gunakan alat AWS IoT SiteWise EMQX CLI di `/greengrass/v2/bin/swe-emqx-cli`

Untuk mengaktifkan otentikasi nama pengguna dan kata sandi melalui Manajemen EMQX menggunakan Linux

1. Ubah kata sandi admin dengan menjalankan perintah berikut:

```
/greengrass/v2/bin/swe-emqx-cli admin change-pwd
```

2. Saat diminta, lakukan hal berikut:

1. Masukkan pengguna administrator Anda saat ini (default adalah `admin`) dan kata sandi (default adalah `public`).
2. Masukkan dan konfirmasikan kata sandi baru Anda.

Jika berhasil, Anda melihat pesan berikut:

```
admin password changed successfully
```

3. Tambahkan pengguna untuk aplikasi eksternal dengan menjalankan perintah berikut:

```
/greengrass/v2/bin/swe-emqx-cli users add
```

4. Saat diminta, lakukan hal berikut:

1. Masukkan nama pengguna untuk pengguna baru.

2. Masukkan dan konfirmasi kata sandi untuk pengguna baru.

Jika berhasil, Anda melihat pesan berikut:

```
User '[username]' created successfully
```

5. Verifikasi konfigurasi pengguna dengan menjalankan perintah berikut:

```
/greengrass/v2/bin/swe-emqx-cli users list
```

Output menunjukkan semua pengguna yang dikonfigurasi:

```
Users:
- [your-added-username]

Total users: 1
```

EMQX Management with Windows

Gunakan alat AWS IoT SiteWise EMQX CLI di salah satu lokasi berikut:

- PowerShell: C:\greengrass\v2\bin\swe-emqx-cli.ps1
- Prompt Perintah: C:\greengrass\v2\bin\swe-emqx-cli.bat

Untuk mengaktifkan otentikasi nama pengguna dan kata sandi melalui Manajemen EMQX menggunakan Windows

1. Ubah kata sandi admin dengan menjalankan perintah berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 admin change-pwd
```

2. Saat diminta, lakukan hal berikut:

1. Masukkan pengguna administrator Anda saat ini (default adalah `admin`) dan kata sandi (default adalah `public`).
2. Masukkan dan konfirmasi kata sandi baru Anda.

Jika berhasil, Anda melihat pesan berikut:

```
admin password changed successfully
```

3. Tambahkan pengguna untuk aplikasi eksternal dengan menjalankan perintah berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users add
```

4. Saat diminta, lakukan hal berikut:

1. Masukkan nama pengguna untuk pengguna baru.
2. Masukkan dan konfirmasi kata sandi untuk pengguna baru.

Jika berhasil, Anda melihat pesan berikut:

```
User '[username]' created successfully
```

5. Verifikasi konfigurasi pengguna dengan menjalankan perintah berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users list
```

Output menunjukkan semua pengguna yang dikonfigurasi:

```
Users:
- [your-added-username]

Total users: 1
```

Hubungkan aplikasi ke broker EMQX di Edge AWS IoT SiteWise

Broker EMQX menggunakan Transport Layer Security (TLS) pada port 8883 untuk mengenkripsi semua komunikasi, memastikan data Anda tetap terlindungi selama transmisi. Bagian ini memandu Anda melalui langkah-langkah untuk membangun koneksi antara aplikasi Anda dan broker EMQX. Mengikuti langkah-langkah ini membantu menjaga integritas dan kerahasiaan data industri Anda. Proses koneksi melibatkan dua pendekatan utama: menggunakan penemuan IP otomatis melalui komponen, atau mengonfigurasi nama DNS dan alamat IP secara manual sebagai Nama Alternatif Subjek (SANs) dalam sertifikat TLS Anda. Setiap metode memiliki kelebihanannya sendiri tergantung

pada pengaturan jaringan dan persyaratan keamanan Anda. Dokumentasi ini akan memandu Anda melalui kedua opsi.

Topik

- [Konfigurasi TLS untuk koneksi aman ke broker EMQX di Edge AWS IoT SiteWise](#)
- [Uji koneksi broker EMQX di Edge AWS IoT SiteWise](#)
- [Gunakan CA Anda sendiri](#)
- [Buka port 8883 untuk koneksi firewall eksternal](#)

Konfigurasi TLS untuk koneksi aman ke broker EMQX di Edge AWS IoT SiteWise

Secara default, AWS IoT Greengrass menghasilkan sertifikat server TLS untuk broker EMQX yang ditandatangani oleh otoritas sertifikat perangkat inti (CA). Untuk informasi selengkapnya, lihat [Menghubungkan perangkat klien ke perangkat AWS IoT Greengrass Core dengan broker MQTT](#).

Ambil sertifikat TLS

Untuk mendapatkan sertifikat CA, jalankan perintah berikut pada host gateway:

Linux

Jalankan perintah berikut dalam sesi shell pada host gateway:

```
/greengrass/v2/bin/swe-emqx-cli cert
```

Perintah ini menampilkan lokasi sertifikat dan mencetak konten sertifikat.

Anda dapat menyimpan sertifikat ke file menggunakan perintah ini:

```
/greengrass/v2/bin/swe-emqx-cli cert --output /path/to/certificate.pem
```

Windows

Jalankan perintah berikut dalam PowerShell sesi di host gateway:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert
```

Perintah ini menampilkan lokasi sertifikat dan mencetak konten sertifikat.

Anda dapat menyimpan sertifikat ke file menggunakan perintah ini:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert --output C:\path\to\certificate.pem
```

CLI secara otomatis menempatkan sertifikat terlepas dari jalur yang tepat pada sistem Anda.

Salin isi file `ca.pem` ke aplikasi eksternal yang Anda sambungkan ke broker. Simpan sebagai `BrokerCoreDeviceCA.pem`.

Tambahkan nama DNS kustom/alamat IP ke sertifikat server TLS

Nama alternatif subjek (SAN) pada sertifikat yang dihasilkan oleh AWS IoT Greengrass adalah `localhost`. Saat membuat koneksi TLS dari luar host gateway, langkah verifikasi TLS gagal karena nama host broker tidak cocok dengan nama host `localhost` pada sertifikat server.

Untuk mengatasi masalah nama host yang tidak cocok, AWS IoT Greengrass sediakan dua cara untuk mengelola titik akhir perangkat inti. Bagian ini mencakup kedua opsi. Untuk informasi selengkapnya, lihat [Mengelola titik akhir perangkat inti](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

- Untuk terhubung ke broker EMQX menggunakan alamat IP perangkat inti, gunakan bagian Penemuan IP Otomatis.
- Untuk terhubung ke broker EMQX menggunakan nama DNS alih-alih alamat IP, Anda menggunakan bagian Manajemen manual.

Automated IP discovery

Opsi ini memungkinkan perangkat inti Anda untuk secara otomatis menemukan alamat IP-nya dan menambahkannya sebagai Nama Alternatif Subjek (SAN) ke sertifikat broker.

1. Tambahkan `aws.greengrass.clientdevices.IPDetector` komponen ke penerapan perangkat inti Anda.
2. Menerapkan perubahan ke perangkat Anda
3. Tunggu hingga penerapan selesai.

Setelah penerapan selesai, Anda dapat membuat koneksi TLS yang aman menggunakan alamat IP broker.

Alamat IP secara otomatis ditambahkan sebagai SAN ke sertifikat broker.

Manual DNS and IP Configuration

Anda dapat menambahkan nama DNS dan alamat IP secara manual sebagai Nama Alternatif Subjek (SANs) ke sertifikat TLS Anda. Metode ini berguna ketika Anda telah mengonfigurasi nama DNS untuk host gateway Anda.

Important

Jika Anda menggunakan IPDetector komponen, hapus dari penerapan Anda sebelum melanjutkan. IPDetectorKomponen mengganti konfigurasi titik akhir manual.

Untuk mengonfigurasi titik akhir secara manual

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway untuk dikonfigurasi.
4. Di bagian konfigurasi gateway Edge, pilih url perangkat inti Greengrass Anda. Halaman perangkat inti muncul.
5. Pilih tab Perangkat klien.
6. Pilih Kelola titik akhir.
7. Di kotak dialog Kelola titik akhir, masukkan nama DNS dan alamat IP apa pun yang ingin Anda tambahkan sebagai. SANs Gunakan port 8883.
8. Pilih Perbarui.

Sertifikat server TLS broker diperbarui secara otomatis untuk menyertakan titik akhir baru Anda.

Untuk memverifikasi pembaruan sertifikat server TLS menggunakan Linux

1. Mulai sesi shell di host gateway Anda.

```
docker exec emqx openssl x509 -in ./data/cert.pem -text -noout | grep -A1  
"Subject Alternative Name"
```

2. Perintah mengembalikan output yang mirip dengan berikut ini:

```
X509v3 Subject Alternative Name:
```

```
DNS:endpoint_you_added, DNS:localhost
```

3. Verifikasi bahwa titik akhir Anda muncul dalam daftar. SANs

Untuk memverifikasi pembaruan sertifikat server TLS menggunakan Windows

1. Mulai sesi shell di host gateway Anda.

```
(Get-PfxCertificate -FilePath "C:\greengrass\v2\work  
\aws.greengrass.clientdevices.mqtt.EMQX\v2\data\cert.pem").Extensions | Where-  
Object { $_.Oid.FriendlyName -eq "Subject Alternative Name" } | ForEach-Object  
{ "Subject Alternative Name:", ($_.Format($true) -split "`n")[0..1] }
```

2. Perintah mengembalikan output yang mirip dengan berikut ini:

```
Subject Alternative Name:  
DNS Name=your-endpoint  
DNS Name=localhost
```

3. Verifikasi bahwa titik akhir yang Anda tambahkan ada dalam daftar. SANs

Uji koneksi broker EMQX di Edge AWS IoT SiteWise

Setelah mengonfigurasi broker EMQX Anda dengan sertifikat TLS dan kredensial otentikasi, penting untuk memverifikasi bahwa pengaturan Anda berfungsi dengan benar. Menguji koneksi membantu memastikan bahwa konfigurasi keamanan Anda diterapkan dengan benar dan klien dapat berhasil membuat koneksi terenkripsi ke broker. Bagian ini menunjukkan cara menguji koneksi broker Anda menggunakan klien antarmuka baris perintah Mosquitto (CLI), alat klien MQTT yang banyak digunakan yang mendukung enkripsi dan otentikasi TLS.

Gunakan klien Mosquitto CLI untuk menguji koneksi broker EMQX

Pada langkah ini kita akan menggunakan klien CLI mosquitto untuk menguji pengaturan kami dan memastikan kami dapat terhubung dengan sukses ke broker menggunakan nama pengguna dan kata sandi yang kami buat sebelumnya. Untuk mendapatkan langkah-langkah BrokerCoreDeviceCA.pem berikut di bawah Langkah 3: Menyiapkan TLS.

```
mosquitto_sub -h hostname/ip address \  
-p 8883 \  
-t "#" \
```

```
-q 1 \  
-u username -P password \  
--cafile BrokerCoreDeviceCA.pem
```

Note

Anda mungkin mendapatkan kesalahan SSL: Verify jika hostname/IP alamat yang Anda sambungkan tidak cocok dengan Nama Alternatif Subjek (SAN) yang ada di sertifikat CA yang Anda berikan ke klien. Lihat “Menambahkan names/IP alamat DNS khusus ke sertifikat server TLS” di bawah Langkah 3: Menyiapkan TLS untuk cara mendapatkan sertifikat dengan SAN yang benar.

Pada titik ini, semua pengguna memiliki akses untuk mempublikasikan dan berlangganan semua topik di broker. Lanjut ke [Siapkan aturan otorisasi untuk AWS IoT SiteWise Edge di EMQX](#).

Gunakan CA Anda sendiri

AWS IoT Greengrass menguraikan cara mengonfigurasi komponen autentikasi perangkat klien Anda sendiri untuk menggunakan otoritas sertifikat (CA) Anda sendiri. Komponen auth perangkat klien (`aws.greengrass.clientdevices.Auth`) mengautentikasi perangkat klien dan mengotorisasi tindakan perangkat klien. Untuk informasi selengkapnya, lihat [Menggunakan otoritas sertifikat Anda sendiri](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Untuk menggunakan CA Anda sendiri, tambahkan `aws.greengrass.clientdevices.Auth` komponen ke penerapan Anda sehingga Anda dapat menentukan konfigurasi kustom.

Buka port 8883 untuk koneksi firewall eksternal

Linux

Dalam aturan firewall host Linux Anda, tambahkan aturan masuk untuk port 8883 untuk memungkinkan koneksi masuk dari luar host gateway. Jika ada firewall di tempat, pastikan bahwa koneksi TLS masuk pada port 8883 diperbolehkan.

Windows

Dalam aturan firewall Microsoft Windows host Anda, tambahkan aturan masuk untuk port 8883 untuk mengizinkan koneksi masuk dari luar host gateway. Pastikan aturannya adalah aturan izinkan, dari jenis port, yang menentukan port 8883. Anda dapat mengonfigurasi ini sesuai

dengan konfigurasi jaringan Anda untuk memungkinkan koneksi dari aplikasi eksternal Anda ke broker.

Siapkan aturan otorisasi untuk AWS IoT SiteWise Edge di EMQX

EMQX mendukung penambahan aturan otorisasi berdasarkan pengidentifikasi seperti nama pengguna, alamat IP atau ID klien. Ini berguna jika Anda ingin membatasi jumlah aplikasi eksternal yang terhubung ke berbagai operasi atau topik.

Topik

- [Konfigurasi otorisasi menggunakan database bawaan dengan Linux](#)
- [Konfigurasi otorisasi menggunakan database bawaan dengan Windows](#)
- [Perbarui konfigurasi penerapan EMQX untuk otorisasi](#)
- [Tambahkan aturan otorisasi melalui Dasbor EMQX untuk pengguna](#)

Konfigurasi otorisasi menggunakan database bawaan dengan Linux

Saat Anda mengonfigurasi aturan otorisasi, ada dua pilihan konfigurasi yang bergantung pada pengaturan penerapan Anda.

- Docker— Jika Anda menjalankan Docker instalasi standar tanpa Litmus Edge, gunakan konfigurasi gateway jembatan Docker. Ini biasanya terjadi ketika Anda hanya menerapkan AWS IoT SiteWise komponen.
- Litmus Edge— Jika Anda telah Litmus Edge menginstal pada gateway Anda, gunakan konfigurasi subnet Litmus Edge jaringan.

 Note

Jika Anda awalnya mengkonfigurasi gateway Docker jembatan dan kemudian menginstal Litmus Edge, konfigurasi ulang aturan otorisasi menggunakan opsi subnet Litmus Edge jaringan untuk memastikan komunikasi yang tepat antara semua komponen.

Untuk menambahkan aturan otorisasi dasar

1. Verifikasi bahwa broker EMQX digunakan dan dijalankan.

2. Mulai sesi shell di host gateway Anda.

Docker without Tepi Lakmus

Untuk Docker instalasi standar tanpa Litmus Edge, jalankan:

```
/greengrass/v2/bin/swe-emqx-cli acl init
```

Litmus Edge network subnet

Jika Anda menggunakan Litmus Edge, tentukan IP subnet jaringan Litmus Edge:

```
docker network inspect LitmusNetwork | grep IPAM -A9
```

Perhatikan nilai Subnet dari output dan jalankan perintah berikut. Ganti `litmus_subnet_ip` dengan nilai Subnet dari langkah sebelumnya.

```
/greengrass/v2/bin/swe-emqx-cli acl init litmus_subnet_ip
```

Alat ini secara otomatis membuat dan menerapkan aturan otorisasi untuk memungkinkan koneksi dari alamat IP yang disediakan ke broker. Ini memungkinkan akses ke semua topik. Ini termasuk kolektor IoT SiteWise OPC UA dan penerbit IoT. SiteWise

3. Lanjut ke [Perbarui konfigurasi penerapan EMQX untuk otorisasi](#).

Konfigurasi otorisasi menggunakan database bawaan dengan Windows

Bagian ini mencakup konfigurasi aturan otorisasi menggunakan database bawaan untuk penerapan Windows.

Untuk menambahkan aturan otorisasi dasar

1. Verifikasi bahwa broker EMQX digunakan dan dijalankan.
2. Jalankan alat AWS IoT SiteWise EMQX CLI:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 acl init
```

Alat ini secara otomatis membuat dan menerapkan aturan ACL yang memungkinkan koneksi dari localhost (127.0.0.1) ke broker. Ini memungkinkan akses ke semua topik. Ini termasuk kolektor IoT SiteWise OPC UA dan penerbit IoT. SiteWise

3. Lanjut ke [Perbarui konfigurasi penerapan EMQX untuk otorisasi](#).

Perbarui konfigurasi penerapan EMQX untuk otorisasi

Untuk memperbarui konfigurasi penerapan EMQX untuk otorisasi

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway untuk dikonfigurasi.
4. Di bagian konfigurasi gateway Edge, salin nilai perangkat inti Greengrass Anda. Simpan untuk digunakan nanti.
5. Buka [konsol AWS IoT](#).
6. Di navigasi kiri, di bawah bagian Kelola, pilih Perangkat Greengrass, lalu Deployment.
7. Temukan nilai perangkat inti yang Anda simpan sebelumnya dan pilih tautan itu untuk membuka penerapan.
8. Pilih tombol dropdown Actions, lalu Revise.
9. Baca pesan yang muncul lalu pilih Revise deployment. Halaman Tentukan target muncul.
10. Pilih Berikutnya hingga Anda mencapai langkah Konfigurasi komponen.
11. Pilih tombol `aws.greengrass.clientdevices.mqtt.EMQX radio`.
12. Pilih tombol Configure component. Halaman konfigurasi muncul untuk komponen.
13. Di bawah Pembaruan konfigurasi, pilih Atur ulang ke konfigurasi default untuk versi komponen: `2.*.*`.
14. Rekatkan konten berikut di bagian Configuration to merge berdasarkan OS Anda.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
```

```
        "type": "built_in_database"
      },
      {
        "type": "file",
        "path": "data/authz/acl.conf"
      }
    ]
  },
  "listeners": {
    "tcp": {
      "default": {
        "enabled": true,
        "enable_authn": false
      }
    },
    "ssl": {
      "default": {
        "enabled": true,
        "enable_authn": true,
        "ssl_options": {
          "verify": "verify_none",
          "fail_if_no_peer_cert": false
        }
      }
    }
  },
  "authentication": {
    "enable": true,
    "backend": "built_in_database",
    "mechanism": "password_based",
    "password_hash_algorithm": {
      "iterations": 210000,
      "mac_fun": "sha512",
      "name": "pbkdf2"
    },
    "user_id_type": "username"
  },
  "dashboard": {
    "listeners": {
      "http": {
        "bind": 18083
      }
    }
  }
}
```



```

    },
    "authMode": "bypass",
    "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883
-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e
EMQX_NODE__NAME=emqx@local",
    "requiresPrivilege": "true"
}

```

Windows

```

{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
          "type": "built_in_database"
        },
        {
          "type": "file",
          "path": "C:\\greengrass\\v2\\work\\
\\aws.greengrass.clientdevices.mqtt.EMQX\\v2\\data\\authz\\acl.conf"
        }
      ]
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    }
  },
}

```

```
    "authentication": {
      "enable": true,
      "backend": "built_in_database",
      "mechanism": "password_based",
      "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
      },
      "user_id_type": "username"
    },
    "dashboard": {
      "listeners": {
        "http": {
          "bind": 18083
        }
      }
    }
  },
  "authMode": "bypass",
  "requiresPrivilege": "true"
}
```

15. Pilih Konfirmasi.
16. Pilih Berikutnya sampai Anda mencapai langkah Review.
17. Pilih Deploy.

Note

Mulai saat ini dan seterusnya, Anda tidak dapat mengedit file ACL untuk memperbarui aturan otorisasi. Atau, Anda dapat melanjutkan ke [Tambahkan aturan otorisasi melalui Dasbor EMQX untuk pengguna](#) setelah penerapan berhasil.

Tambahkan aturan otorisasi melalui Dasbor EMQX untuk pengguna

Anda dapat menambahkan atau memperbarui aturan otorisasi menggunakan Dasbor EMQX atau alat EMQX AWS IoT SiteWise CLI. Alat AWS IoT SiteWise CLI EMQX mengelola otorisasi menggunakan database bawaan EMQX.

Note

Menambahkan aturan otorisasi adalah langkah konfigurasi lanjutan yang memerlukan pemahaman tentang pola topik MQTT dan kontrol akses. Untuk informasi selengkapnya tentang membuat aturan otorisasi menggunakan database bawaan EMQX, lihat [Menggunakan Basis Data Bawaan](#) di Dokumen EMQX.

Note

Instruksi terkait EMQX yang disediakan hanya untuk referensi. Karena dokumentasi dan fitur EMQX dapat berubah seiring waktu, dan kami tidak memelihara dokumentasinya, kami sarankan untuk berkonsultasi dengan [dokumentasi resmi EMQX untuk informasi](#) terbaru.

EMQX dashboard

Prosedur ini menunjukkan bagaimana Anda dapat menambahkan aturan otorisasi di dasbor EMQX.

Dasbor EMQX hanya dapat diakses dari dalam host gateway. Jika Anda mencoba terhubung dari luar host gateway, Anda tidak dapat mengakses dasbor.

Untuk menambahkan aturan otorisasi menggunakan Dasbor EMQX

1. Pastikan Anda berada di dalam host gateway.
2. Buka jendela browser dan kunjungi <http://localhost:18083/>.
3. Masuk ke dasbor EMQX. Prosedur ini mengasumsikan bahwa Anda telah mengubah kredensi login default Anda menjadi sesuatu yang Anda pilih. Untuk informasi lebih lanjut tentang penyiapan awal, lihat [Aktifkan otentikasi nama pengguna dan kata sandi](#).
4. Pilih ikon perisai, lalu Otorisasi dari menu tarik-turun.
5. Pilih tombol Permissions pada baris Built-in Database.
6. Di bagian otorisasi Basis Data bawaan, tambahkan atau perbarui aturan otorisasi pengguna untuk kebutuhan bisnis Anda. Untuk panduan selengkapnya tentang membuat aturan, lihat bagian [Use Built-in Database](#) di Dokumen EMQX.

AWS IoT SiteWise CLI tool using Linux

Untuk mengelola aturan otorisasi menggunakan alat AWS IoT SiteWise EMQX CLI di Linux:

- Tambahkan aturan otorisasi untuk pengguna menggunakan format berikut:

```
/greengrass/v2/bin/swe-emqx-cli auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example Menambahkan aturan otorisasi untuk pengguna

Contoh ini menunjukkan cara menambahkan aturan untuk pengguna bernama `system1`:

```
/greengrass/v2/bin/swe-emqx-cli auth add system1 \  
  publish allow "sensors/#" \  
  subscribe allow "control/#" \  
  all deny "#"
```

Example : Lihat aturan otorisasi untuk pengguna

Untuk melihat aturan otorisasi bagi `system1` pengguna, jalankan perintah berikut:

```
/greengrass/v2/bin/swe-emqx-cli auth list system1
```

Example : Lihat semua aturan otorisasi yang ada

Untuk melihat semua aturan otorisasi yang Anda miliki saat ini, jalankan perintah berikut:

```
/greengrass/v2/bin/swe-emqx-cli auth list
```

Example : Hapus semua aturan otorisasi untuk pengguna

Untuk menghapus semua aturan otorisasi yang diterapkan pada pengguna tertentu, jalankan perintah berikut:

```
/greengrass/v2/bin/swe-emqx-cli auth delete system1
```

Anda diminta untuk mengkonfirmasi penghapusan.

AWS IoT SiteWise CLI tool using Windows

Untuk mengelola aturan otorisasi menggunakan alat AWS IoT SiteWise EMQX CLI di: Windows PowerShell

- Tambahkan aturan otorisasi untuk pengguna menggunakan format berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example : Tambahkan aturan otorisasi untuk pengguna

Contoh ini menunjukkan cara menambahkan aturan untuk pengguna bernama `system1`:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add system1 `
  publish allow "sensors/#" `
  subscribe allow "control/#" `
  all deny "#"
```

Example : Lihat aturan otorisasi untuk pengguna

Untuk melihat aturan otorisasi bagi `system1` pengguna, jalankan perintah berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list system1
```

Example : Lihat semua aturan otorisasi yang ada

Untuk melihat semua aturan otorisasi yang Anda miliki saat ini, jalankan perintah berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list
```

Example : Hapus semua aturan otorisasi untuk pengguna

Untuk menghapus semua aturan otorisasi yang diterapkan pada pengguna tertentu, jalankan perintah berikut:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth delete system1
```

Anda diminta untuk mengkonfirmasi penghapusan.

Memproses dan memvisualisasikan data dengan SiteWise Edge dan alat sumber terbuka

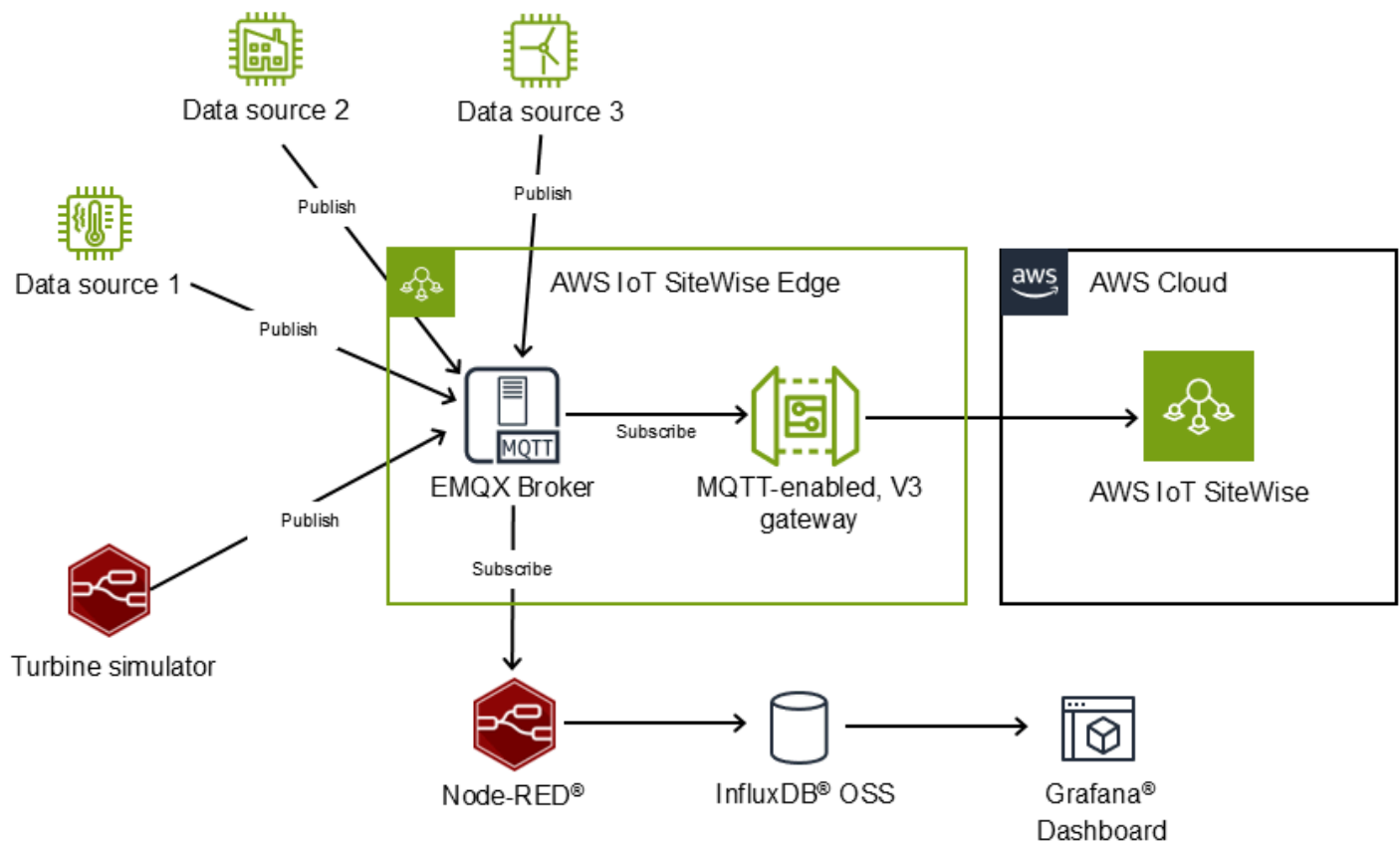
Konfigurasi gateway berkemampuan AWS IoT SiteWise Edge MQTT dengan alat sumber terbuka untuk pemrosesan dan visualisasi lokal guna meningkatkan kemampuan manajemen data industri Anda.

Dengan SiteWise Edge, Anda dapat membuat pipeline pemrosesan data lokal menggunakan alat sumber terbuka eksternal. [Gunakan Node-red® untuk menyimpan data deret waktu dengan InfluxDB®, dan memantau operasi melalui dasbor Grafana®.](#)

Node-red memproses dan mengubah aliran data Anda, sementara InfluxDB menyediakan penyimpanan data deret waktu. Grafana menampilkan data operasional real-time Anda. Gunakan alat ini dengan SiteWise Edge untuk menyinkronkan data antara lingkungan lokal Anda dan lingkungan AWS Cloud, memberi Anda wawasan lokal langsung dan kemampuan analitik berbasis cloud jangka panjang.

Note

Node-red®, InfluxDB®, dan Grafana® bukan vendor atau pemasok untuk Edge. SiteWise



Note

Dalam panduan ini, kami menggunakan Grafana SiteWise for Edge versi open-source sebagai lawan dari layanan [Grafana](#) yang [Dikelola Amazon](#).

Opsi deployment

Anda dapat menerapkan solusi ini menggunakan salah satu dari dua pendekatan. Dengan pengaturan Microsoft Windows manual, Anda mengontrol konfigurasi komponen dan integrasi dengan infrastruktur Anda. Dengan Linux, Anda dapat menggunakan Docker untuk menyebarkan komponen yang telah dikonfigurasi sebelumnya dalam wadah.

Pilih metode yang memenuhi persyaratan operasional Anda.

- [Mengatur integrasi open source secara manual \(Windows\)](#)— Untuk konfigurasi kustom atau infrastruktur yang ada
- [Siapkan integrasi sumber terbuka dengan Docker \(Linux\)](#)— Untuk penyebaran cepat dengan komponen yang telah dikonfigurasi sebelumnya

Ikhtisar contoh peternakan angin

Panduan ini menggunakan contoh ladang angin untuk menunjukkan bagaimana Anda dapat memantau kecepatan angin untuk turbin di ladang angin. Skenario praktis ini menggambarkan kebutuhan pemantauan industri umum di mana visibilitas lokal dan berbasis cloud berharga untuk efisiensi operasional.

Dengan integrasi ini, Anda dapat:

- Kumpulkan data dari peralatan industri menggunakan gateway AWS IoT SiteWise Edge
- Memproses data secara lokal dengan Node-Red, InfluxDB, dan Grafana
- Menyimpan data secara lokal menggunakan InfluxDB
- Pantau data secara real time menggunakan dasbor Grafana

Sepanjang panduan ini, kami menggunakan contoh ladang angin. Kami menggunakan Node-red untuk mensimulasikan turbin yang menghasilkan data kecepatan angin. Node-red menerjemahkan payload data, menerbitkan data ke broker SiteWise Edge MQTT, berlangganan untuk menerima data dari broker, dan menyimpan data secara lokal di InfluxDB. Pendekatan ini memastikan bahwa semua data operasional tersedia baik secara lokal untuk akses langsung dan di cloud untuk analisis lebih lanjut. Dengan menerapkan pola ini, Anda mendapatkan ketahanan terhadap gangguan jaringan sambil mempertahankan kemampuan untuk melakukan analitik lanjutan di AWS Cloud. Grafana terhubung ke InfluxDB untuk pemantauan lokal, memberikan operator visibilitas real-time ke dalam metrik tanpa dependensi cloud. Gateway berkemampuan SiteWise Edge MQTT terhubung ke broker MQTT yang sama untuk mengirim data AWS IoT SiteWise, menciptakan jembatan antara operasi edge Anda dan layanan berbasis cloud.

Anda dapat menggunakan data dan konfigurasi Anda sendiri untuk membuat alur kerja serupa yang disesuaikan dengan kebutuhan industri spesifik Anda, baik Anda memantau peralatan manufaktur, infrastruktur utilitas, atau aset industri lainnya.

Persyaratan untuk integrasi sumber terbuka

Sebelum menerapkan integrasi sumber terbuka dengan SiteWise Edge, pastikan lingkungan Anda memenuhi persyaratan yang diperlukan.

- Persyaratan perangkat keras - Perangkat keras gateway Anda harus memenuhi persyaratan untuk gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise Persyaratan gateway yang dihosting sendiri di Edge](#) gateway V3 berkemampuan MQTT, dan [Persyaratan untuk aplikasi AWS IoT SiteWise Edge](#)

 Important

Saat menerapkan komponen sumber terbuka tambahan, pastikan perangkat keras Anda memenuhi persyaratan untuk InfluxDB, Node-Red, dan Grafana.

- Konfigurasi jaringan Anda harus mendukung komunikasi lokal antara komponen dan konektivitas cloud untuk SiteWise Edge.
- Semua layanan harus berjalan pada host yang sama.

Pertimbangan keamanan

Kami menyarankan Anda mengenkripsi semua komunikasi antar komponen, terutama saat mengakses antarmuka dari jaringan non-lokal. Terapkan kontrol akses yang tepat untuk setiap komponen dan ikuti praktik AWS terbaik untuk konfigurasi gateway AWS IoT SiteWise Edge dan keamanan AWS akun.

Lingkungan pengembangan

Panduan ini menunjukkan Node-Red, InfluxDB, dan Grafana berjalan dan diakses secara lokal di host gateway. Untuk penerapan produksi yang memerlukan akses eksternal, terapkan langkah-langkah keamanan termasuk enkripsi TLS, otentikasi, dan otorisasi. Ikuti praktik terbaik keamanan setiap aplikasi.

Perangkat lunak pihak ketiga

Solusi ini menggunakan perangkat lunak pihak ketiga yang tidak dikelola oleh AWS, termasuk InfluxDB, Node-red, Grafana, dan plugin. `node-red-contrib-influxdb` Sebelum penerapan, pastikan komponen ini mematuhi persyaratan keamanan, standar kepatuhan, dan kebijakan tata kelola organisasi Anda.

 Important

Panduan ini merujuk dan menggunakan perangkat lunak pihak ketiga yang tidak dimiliki atau dikelola oleh AWS. Sebelum implementasi, pastikan bahwa semua komponen memenuhi persyaratan keamanan, kepatuhan, dan tata kelola Anda. Tetap perbarui semua perangkat lunak dengan tambalan keamanan terbaru dan ikuti praktik terbaik untuk mengamankan penerapan edge Anda.

InfluxDB, Node-red, Grafana bukan vendor atau pemasok untuk Edge. SiteWise

Pertimbangan lainnya

Pertimbangkan faktor-faktor tambahan ini saat menerapkan integrasi sumber terbuka dengan SiteWise Edge.

- Gunakan versi terbaru dari semua layanan, alat, dan komponen.
- Filter dan agregat data secara lokal sebelum transmisi cloud untuk mengurangi biaya konsumsi AWS IoT SiteWise data. Konfigurasi periode retensi data yang sesuai di InfluxDB dan ukur perangkat keras gateway Anda dengan benar. Untuk informasi selengkapnya, lihat [harga AWS IoT SiteWise](#).
- Menerapkan prosedur pencadangan reguler untuk semua data.
- Pantau penggunaan sumber daya di gateway Anda dan konfigurasi batas sumber daya yang sesuai untuk setiap komponen. Menerapkan kebijakan retensi data di InfluxDB untuk mengelola penggunaan disk.

Mengatur integrasi open source secara manual (Windows)

Gunakan panduan ini untuk membuat bucket deret waktu secara manual untuk data kecepatan angin yang terhubung dengan Grafana® dan Node-red®.

Instal dan konfigurasi Node-Red, InfluxDB®, dan Grafana secara manual untuk mengontrol konfigurasi penerapan Anda. Microsoft Windows Anda dapat menyimpan dan mengelola data deret waktu dari perangkat Anda menggunakan InfluxDB.

Prasyarat pengaturan manual

Sebelum Anda mulai, lengkapi persyaratan ini:

Note

Jalankan semua layanan (SiteWise Edge, InfluxDB, Node-red, dan Grafana) pada host yang sama.

- Instal gateway V3 berkemampuan MQTT. Untuk informasi selengkapnya, lihat [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#).

- Instal dan jalankan layanan ini secara lokal:
 - InfluxDB OSS v2. Untuk langkah-langkah instalasi, lihat [Menginstal InfluxDB](#).
 - Simpul-merah. Untuk langkah-langkah instalasi, lihat [Menginstal Node-merah secara lokal](#).
 - Grafana. Untuk langkah-langkah instalasi, lihat [Menginstal Grafana](#).

Siapkan penyimpanan lokal dengan InfluxDB

Dengan InfluxDB®, Anda dapat menyimpan data deret waktu dari perangkat Anda secara lokal. Tujuan dari kemampuan penyimpanan lokal adalah untuk menjaga visibilitas operasional selama gangguan jaringan dan mengurangi latensi untuk aplikasi yang kritis waktu. Anda dapat melakukan analisis dan visualisasi di tepi sambil tetap memiliki opsi untuk meneruskan data secara selektif ke cloud.

Di bagian ini, Anda membuat bucket deret waktu untuk data kecepatan angin turbin dan menghasilkan token API untuk konektivitas Grafana® dan Node-red®. Bucket InfluxDB berfungsi sebagai wadah penyimpanan khusus untuk data deret waktu Anda, mirip dengan database dalam sistem tradisional. Token API memungkinkan akses terprogram yang aman ke data Anda.

Untuk mengatur InfluxDB

1. [Setelah menyelesaikan langkah-langkah prasyarat dan memastikan semua alat berjalan pada host yang sama, buka browser web Anda dan buka `http://127.0.0.1:8086`.](#)
2. (Opsional) Aktifkan enkripsi TLS untuk keamanan yang ditingkatkan. Untuk informasi selengkapnya, lihat [Mengaktifkan enkripsi TLS](#) di InfluxData Documentation
3. Buat bucket InfluxDB time series untuk menyimpan data dari Node-Red. Bucket akan berfungsi sebagai wadah khusus untuk data ladang angin Anda, memungkinkan Anda mengatur dan mengelola kebijakan retensi khusus untuk kumpulan data ini. Untuk informasi selengkapnya, lihat [Mengelola bucket](#) di InfluxData Documentation
4. (Opsional) Konfigurasi periode retensi data untuk lokasi tepi Anda. Menyetel periode retensi yang sesuai membantu mengelola sumber daya penyimpanan secara efisien dengan secara otomatis menghapus data lama yang tidak lagi diperlukan untuk operasi lokal.

Untuk informasi tentang retensi data, lihat [Retensi data di InfluxDB](#) di InfluxData Documentation

5. Hasilkan token API untuk bucket. Token ini akan memungkinkan komunikasi yang aman antara InfluxDB dan komponen lain seperti Node-Red dan Grafana. Dengan cara ini, hanya layanan resmi yang dapat membaca dari atau menulis ke penyimpanan data Anda. Untuk informasi selengkapnya, lihat [Membuat token](#) di InfluxData Documentation.

Setelah Anda menyelesaikan langkah-langkah ini, Anda dapat menyimpan data deret waktu di instans InfluxDB Anda, memberikan dasar untuk persistensi dan analisis data lokal di lingkungan edge Anda.

Konfigurasi alur Node-merah untuk integrasi data AWS IoT SiteWise

Dengan Node-Red®, Anda dapat menerapkan dua aliran untuk mengelola data antara perangkat Anda dan AWS IoT SiteWise. Alur ini bekerja sama untuk menciptakan solusi manajemen data komprehensif yang membahas aliran data lokal dan cloud.

- Aliran publikasi data — Menerbitkan ke cloud. Aliran publikasi data mengirimkan data ke AWS IoT SiteWise. Aliran ini mensimulasikan perangkat turbin dengan menghasilkan data sensor, menerjemahkannya ke AWS IoT SiteWise format, dan menerbitkan ke broker SiteWise Edge MQTT. Ini memungkinkan Anda memanfaatkan AWS IoT SiteWise kemampuan cloud untuk penyimpanan, analitik, dan integrasi dengan AWS layanan lain.

Untuk informasi selengkapnya, lihat [Konfigurasi alur publikasi data](#).

- Aliran retensi data — Menyimpan data di tepi. Alur retensi data berlangganan broker SiteWise Edge MQTT untuk menerima data, menerjemahkannya ke dalam format InfluxDB®, dan menyimpannya secara lokal untuk pemantauan. Penyimpanan lokal ini menyediakan akses langsung ke data operasional, mengurangi latensi untuk aplikasi kritis waktu, dan memastikan kontinuitas selama gangguan jaringan.

Untuk informasi selengkapnya, lihat [Konfigurasi aliran retensi data](#).

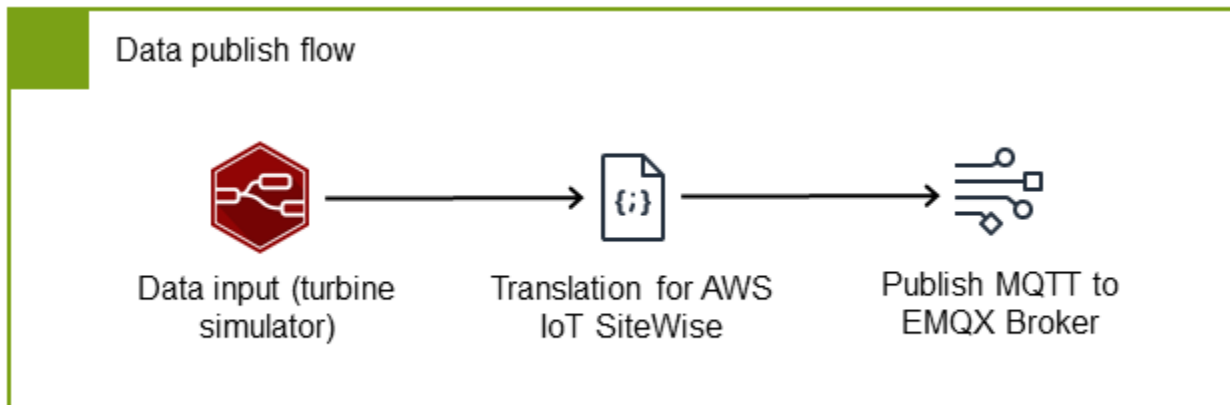
Kedua aliran ini bekerja sama untuk memastikan data dikirim AWS IoT SiteWise dan disimpan secara lokal untuk akses langsung.

[Untuk mengakses konsol Node-red Anda, buka `http://127.0.0.1:1880`](#). Untuk informasi tentang mengaktifkan TLS, lihat [Mengaktifkan enkripsi TLS](#).

Konfigurasi alur publikasi data

Aliran publikasi data menggunakan tiga node untuk membuat pipeline yang mengirimkan data industri Anda ke cloud. Alur ini penting untuk memungkinkan analitik berbasis cloud, penyimpanan jangka panjang, dan integrasi dengan layanan lain. AWS Pertama, data perangkat simulasi dikirim ke broker SiteWise Edge MQTT. Gateway mengambil data dari broker yang memungkinkan transmisi ke AWS IoT SiteWise cloud, di mana Anda dapat memanfaatkan kemampuan analitik dan visualisasi yang kuat.

- Input data - Menerima data perangkat dari peralatan industri atau simulator Anda
- Penerjemah data untuk AWS IoT SiteWise - Menerjemahkan data ke AWS IoT SiteWise format untuk memastikan kompatibilitas dengan gateway SiteWise Edge
- Penerbit MQTT - Menerbitkan data ke broker SiteWise Edge MQTT, membuatnya tersedia untuk konsumen lokal dan cloud



Konfigurasi simpul input data

Dalam contoh ini, simpul input data menggunakan perangkat turbin angin simulasi yang menghasilkan data kecepatan angin. Node ini berfungsi sebagai titik masuk untuk data industri Anda, apakah itu berasal dari sumber simulasi (seperti dalam contoh kami) atau dari peralatan industri aktual di lingkungan produksi.

Kami menggunakan format JSON khusus untuk muatan data guna menyediakan struktur standar yang bekerja secara efisien dengan alat pemrosesan lokal dan layanan cloud. AWS IoT SiteWise Format ini mencakup metadata penting seperti stempel waktu dan indikator kualitas di samping nilai pengukuran aktual, memungkinkan manajemen data yang komprehensif dan pelacakan kualitas di seluruh pipeline Anda. Impor node injeksi untuk menerima data simulasi dalam format JSON standar ini dengan stempel waktu, indikator kualitas, dan nilai.

Untuk informasi selengkapnya tentang node-Red inject node, lihat bagian [Inject di Dokumentasi Node-RED](#).

Simulator turbin menghasilkan data kecepatan angin setiap detik dalam format JSON standar ini:

Example : Muatan data turbin

```
{
  name: string,          // Property name/identifier
  timestamp: number,     // Epoch time in nanoseconds
  quality: "GOOD" | "UNCERTAIN" | "BAD",
  value: number | string | boolean
}
```

Format ini memberikan beberapa manfaat:

- **nameBidang** mengidentifikasi properti atau pengukuran tertentu, memungkinkan Anda melacak beberapa titik data dari perangkat yang sama
- **timestamp** Dalam nanodetik memastikan pelacakan waktu yang tepat untuk analisis historis yang akurat
- **qualityIndikator** membantu Anda memfilter dan mengelola data berdasarkan keandalannya
- **valueBidang** fleksibel mendukung tipe data yang berbeda untuk mengakomodasi berbagai output sensor

Example : Menyuntikkan simpul simulator turbin

```
[
  {
    "id": "string",
    "type": "inject",
    "z": "string",
    "name": "Turbine Simulator",
    "props": [
      {
        "p": "payload.timestamp",
        "v": "",
        "vt": "date"
      },
      {
        "p": "payload.quality",
        "v": "GOOD",
        "vt": "str"
      },
      {
        "p": "payload.value",
```

```

        "v": "$random()",
        "vt": "jsonata"
    },
    {
        "p": "payload.name",
        "v": "/Renton/WindFarm/Turbine/WindSpeed",
        "vt": "str"
    }
],
"repeat": "1",
"crontab": "",
"once": false,
"onceDelay": "",
"topic": "",
"x": 270,
"y": 200,
"wires": [
    [
        "string"
    ]
]
}
]
```

Konfigurasi node untuk terjemahan data

Gateway SiteWise Edge membutuhkan data dalam format tertentu untuk memastikan kompatibilitas dengan AWS IoT SiteWise cloud. Node penerjemah adalah komponen penting yang mengubah data input Anda ke format AWS IoT SiteWise payload yang diperlukan. Langkah penerjemahan ini memastikan bahwa data industri Anda dapat diproses, disimpan, dan kemudian dianalisis dengan benar di lingkungan AWS IoT SiteWise cloud.

Dengan menstandarisasi format data pada tahap ini, Anda mengaktifkan integrasi antara perangkat edge dan layanan cloud tempat Anda dapat menggunakan pemodelan aset, analitik, dan kemampuan visualisasi. Gunakan struktur ini:

Example : Struktur muatan untuk penguraian data SiteWise Edge

```

{
  "propertyAlias": "string",
  "propertyValues": [
    {
```

```

    "value": {
      "booleanValue": boolean,
      "doubleValue": number,
      "integerValue": number,
      "stringValue": "string"
    },
    "timestamp": {
      "timeInSeconds": number,
      "offsetInNanos": number
    },
    "quality": "GOOD" | "UNCERTAIN" | "BAD",
  }
}

```

Note

propertyAliasCocokkan dengan hierarki topik MQTT Anda (misalnya, /Renton/WindFarm/Turbine/WindSpeed Ini memastikan bahwa data Anda terkait dengan properti aset yang benar di AWS IoT SiteWise. Untuk informasi selengkapnya, lihat konsep “Alias aliran data” di [AWS IoT SiteWise konsep](#).

1. Impor node fungsi contoh untuk terjemahan AWS IoT SiteWise payload. Fungsi ini menangani konversi dari format input standar Anda ke format AWS IoT SiteWise-kompatibel, memastikan pemformatan stempel waktu yang tepat, indikator kualitas, dan pengetikan nilai.

```

[
  {
    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to SiteWise payload",
    "func": "let input = msg.payload;\nlet output = {};\n\n\noutput[\"propertyAlias\"] = input.name;\n\n\nlet propertyVal = {}\n\n\nlet\n  timeInSeconds = Math.floor(input.timestamp / 1000);\nlet offsetInNanos =\n  (input.timestamp % 1000) * 1000000;\n\n\npropertyVal[\"timestamp\"] = {\n  \"timeInSeconds\": timeInSeconds,\n  \"offsetInNanos\": offsetInNanos,\n};\n\n\npropertyVal[\"quality\"] = input.quality\n\n\nlet typeNameConverter = {\n  \"number\": (x) => Number.isInteger(x) ? \"integerValue\" : \"doubleValue\",\n  \"boolean\": (x) => \"booleanValue\", \n  \"string\": (x) =>\n  \"stringValue\", \n}\n\nlet typeName = typeNameConverter[typeof input.value]

```



```
(input.value)\npropertyVal[\"value\"] = {}\npropertyVal[\"value\"][typeName]
= input.value;\n\noutput[\"propertyValues\"] = [propertyVal]\n\nreturn {\n
payload: JSON.stringify(output)\n};",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 530,
  "y": 200,
  "wires": [
    [
      "string"
    ]
  ]
}
```

2. Verifikasi bahwa JavaScript kode menerjemahkan data kecepatan angin dengan benar. Fungsi ini melakukan beberapa tugas penting:
 - Mengekstrak nama properti dari input dan menetakannya sebagai PropertyAlias
 - Mengonversi stempel waktu dari milidetik ke format detik dan nanodetik yang diperlukan
 - Mempertahankan indikator kualitas untuk pelacakan keandalan data
 - Secara otomatis mendeteksi jenis nilai dan memformatnya sesuai dengan persyaratan AWS IoT SiteWise
3. Hubungkan node ke alur Anda, hubungkan antara node input data dan penerbit MQTT.

Untuk panduan tentang menulis fungsi khusus untuk kebutuhan bisnis Anda, lihat [Fungsi Menulis](#) dalam Node-RED Dokumentasi

Konfigurasi penerbit MQTT

Setelah terjemahan, data siap untuk dipublikasikan ke broker SiteWise Edge MQTT.

Konfigurasi penerbit MQTT dengan pengaturan ini untuk mengirim data ke broker Edge MQTT: SiteWise

Untuk mengimpor node keluar MQTT

1. Impor node konfigurasi MQTT keluar menggunakan. "type": "mqtt out" Node keluar MQTT memungkinkan Anda berbagi konfigurasi broker.
2. Masukkan pasangan nilai kunci untuk informasi yang relevan dengan koneksi broker MQTT dan perutean pesan.

Impor mqtt out simulasi contoh.

Example

```
[
  {
    "id": "string",
    "type": "mqtt out",
    "z": "string",
    "name": "Publish to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "string",
    "x": 830,
    "y": 200,
    "wires": []
  },
  {
    "id": "string",
    "type": "mqtt-broker",
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientId": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
```

```
    "autoUnsubscribe": true,  
    "birthTopic": "",  
    "birthQos": "0",  
    "birthPayload": "",  
    "birthMsg": {},  
    "closeTopic": "",  
    "closePayload": "",  
    "closeMsg": {},  
    "willTopic": "",  
    "willQos": "0",  
    "willPayload": "",  
    "willMsg": {},  
    "userProps": "",  
    "sessionExpiry": ""  
  }  
]
```

Contoh MQTT out node menciptakan koneksi MQTT dengan informasi berikut:

- Server: 127.0.0.1
- Pelabuhan: 1883
- Protokol: MQTT V5

Kemudian, node keluar MQTT mengonfigurasi perutean pesan dengan informasi berikut:

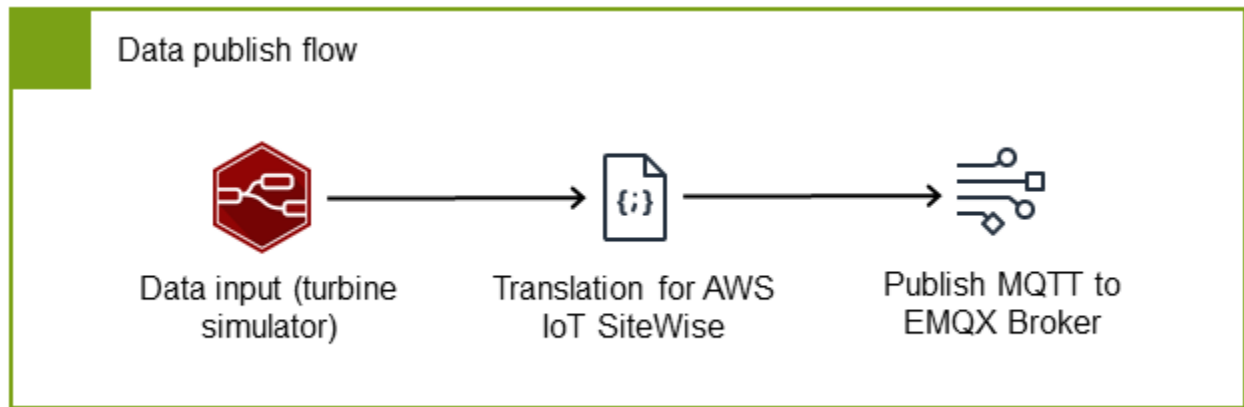
- Topik: /Renton/WindFarm/Turbine/WindSpeed
- QoS: 1

Menyebarkan dan memverifikasi node

Setelah mengonfigurasi tiga node aliran publikasi data, ikuti langkah-langkah ini untuk menyebarkan aliran dan memverifikasi bahwa data sedang dikirim dengan benar AWS IoT SiteWise

Untuk menyebarkan dan memverifikasi koneksi

1. Connect ketiga node seperti yang ditunjukkan dalam aliran data publish.



2. Pilih Deploy untuk menerapkan semua perubahan koneksi node.
3. Arahkan ke [AWS IoT SiteWise konsol](#) dan pilih Aliran data.
4. Pastikan awalan Alias dipilih di menu dropdown. Kemudian, cari /Renton/WindFarm/Turbine/WindSpeed alias.

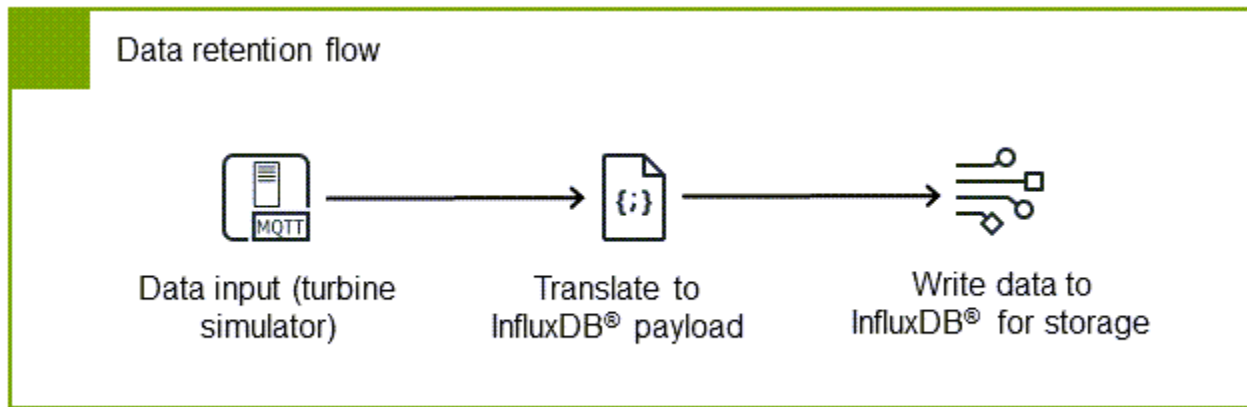
Jika Anda melihat alias yang benar dalam pencarian Anda, Anda telah menerapkan aliran dan transmisi data yang diverifikasi.

Konfigurasi aliran retensi data

Aliran retensi data dapat digunakan untuk menjaga visibilitas operasional di tepi. Ini berguna selama gangguan jaringan atau ketika Anda membutuhkan akses langsung ke data Anda. Alur ini berlangganan broker MQTT untuk menerima data perangkat, mengubahnya menjadi format InfluxDB®, dan menyimpannya secara lokal. Dengan menerapkan alur ini, Anda membuat penyimpanan data lokal yang tangguh yang dapat diakses operator tanpa dependensi cloud, memungkinkan pemantauan waktu nyata dan pengambilan keputusan di tepi.

Alur terdiri dari tiga komponen utama yang bekerja sama untuk memastikan data Anda ditangkap dan disimpan dengan benar:

- Klien berlangganan MQTT - Menerima data dari broker, memastikan Anda menangkap semua data industri yang relevan
- Penerjemah InfluxDB - Mengonversi AWS IoT SiteWise payload ke format InfluxDB, menyiapkan data untuk penyimpanan deret waktu yang efisien
- Penulis InfluxDB - Menangani penyimpanan lokal, memastikan persistensi data dan ketersediaan untuk aplikasi lokal



Siapkan klien berlangganan MQTT

- Konfigurasi klien berlangganan MQTT di Node-Red untuk menerima data dari broker MQTT EMQX dengan mengimpor contoh di bawah ini. AWS IoT SiteWise

Example : MQTT di simpul

```
[
  {
    "id": "string",
    "type": "mqtt in",
    "z": "string",
    "name": "Subscribe to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "datatype": "auto-detect",
    "broker": "string",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 290,
    "y": 340,
    "wires": [
      [
        "string"
      ]
    ]
  },
  {
```

```
"id": "string",
"type": "mqtt-broker",
"name": "emqx",
"broker": "127.0.0.1",
"port": "1883",
"clientId": "",
"autoConnect": true,
"usetls": false,
"protocolVersion": "5",
"keepalive": 15,
"cleansession": true,
"autoUnsubscribe": true,
"birthTopic": "",
"birthQos": "0",
"birthPayload": "",
"birthMsg": {},
"closeTopic": "",
"closePayload": "",
"closeMsg": {},
"willTopic": "",
"willQos": "0",
"willPayload": "",
"willMsg": {},
"userProps": "",
"sessionExpiry": ""
}
]
```

Langganan ini memastikan bahwa semua data relevan yang dipublikasikan ke broker ditangkap untuk penyimpanan lokal, memberikan catatan lengkap operasi industri Anda. Node menggunakan parameter koneksi MQTT yang sama dengan [Konfigurasi penerbit MQTT](#) bagian, dengan pengaturan langganan berikut:

- Topik — /Renton/WindFarm/Turbine/WindSpeed
- QoS — 1

Untuk informasi selengkapnya, lihat [Connect to an MQTT Broker](#) di Dokumentasi. Node-RED

Konfigurasi penerjemah InfluxDB

[InfluxDB mengatur data menggunakan tag untuk pengindeksan dan bidang untuk nilai.](#) Organisasi ini mengoptimalkan kinerja kueri dan efisiensi penyimpanan untuk data deret waktu. Impor node fungsi contoh yang berisi JavaScript kode untuk mengonversi AWS IoT SiteWise payload ke format InfluxDB. Penerjemah membagi properti menjadi dua kelompok:

- Tag - Kualitas dan properti nama untuk pengindeksan yang efisien
- Bidang - Timestamp (dalam milidetik sejak zaman) dan nilai

Example : Fungsi simpul menerjemahkan ke muatan InfluxDB

```
[
  {
    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to InfluxDB payload",
    "func": "let data = msg.payload;\n\nlet timeInSeconds =
data.propertyValues[0].timestamp.timeInSeconds;\nlet offsetInNanos =
data.propertyValues[0].timestamp.offsetInNanos;\nlet timestampInMilliseconds =
(timeInSeconds * 1000) + (offsetInNanos / 1000000);\n\nmsg.payload = [\n  {\n    \"timestamp(milliseconds_since_epoch)\": timestampInMilliseconds,\n    \"value\": data.propertyValues[0].value.doubleValue\n  },\n  {\n    \"name\": data.propertyAlias,\n    \"quality\": data.propertyValues[0].quality\n  }\n]\n\nreturn msg",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 560,
    "y": 340,
    "wires": [
      [
        "string"
      ]
    ]
  }
]
```

Untuk opsi konfigurasi tambahan, lihat [node-red-contrib-influxdb](#) di repositori Node-Red GitHub .

Siapkan penulis InfluxDB

Node penulis InfluxDB adalah komponen terakhir dalam aliran retensi data Anda, yang bertanggung jawab untuk menyimpan data industri Anda di database InfluxDB lokal. Penyimpanan lokal ini penting untuk menjaga visibilitas operasional selama gangguan jaringan dan menyediakan akses langsung ke data untuk aplikasi kritis waktu.

1. Instal `node-red-contrib-influxdb` paket melalui opsi Kelola palet. Paket ini menyediakan node yang diperlukan untuk menghubungkan Node-Red dengan InfluxDB.
2. Tambahkan node keluar InfluxDB ke alur Anda. Node ini akan menangani penulisan data aktual ke database InfluxDB Anda.
3. Konfigurasikan properti server untuk membuat koneksi aman ke instans InfluxDB Anda:
 - a. Set Version to 2.0 - Ini menentukan bahwa Anda terhubung ke InfluxDB v2.x, yang menggunakan API yang berbeda dari versi sebelumnya
 - b. Setel URL ke `http://127.0.0.1:8086` - Ini menunjuk ke instans InfluxDB lokal Anda
 - c. Masukkan token otentikasi InfluxDB Anda. Token aman ini mengotorisasi koneksi ke database Anda. Anda membuat token selama [Siapkan penyimpanan lokal dengan InfluxDB](#) prosedur.
4. Tentukan parameter lokasi penyimpanan untuk menentukan di mana dan bagaimana data Anda akan disimpan:
 - a. Masukkan nama Organisasi InfluxDB Anda — Organisasi adalah ruang kerja untuk sekelompok pengguna, tempat bucket dan dasbor Anda berada. Untuk informasi selengkapnya, lihat [Mengelola organisasi](#) di InfluxData Documentation.
 - b. Tentukan Bucket InfluxDB (misalnya, `WindFarmData`) - Bucket ini setara dengan database dalam sistem tradisional, berfungsi sebagai wadah untuk data deret waktu Anda
 - c. Mengatur Pengukuran InfluxDB (misalnya, `TurbineData`) - Pengukurannya mirip dengan tabel dalam database relasional, mengatur titik data terkait

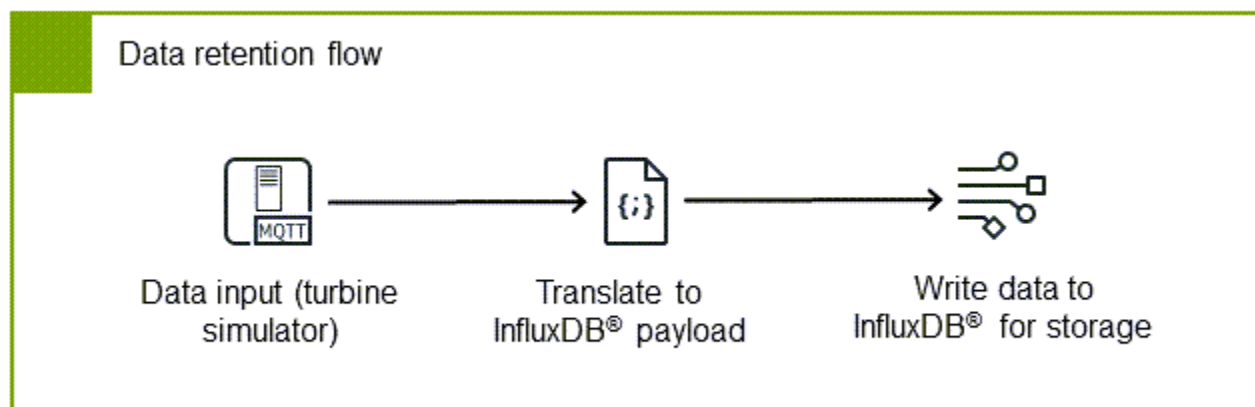
Note

Temukan nama organisasi Anda di sidebar kiri instans InfluxDB. Konsep organisasi, bucket, dan pengukuran sangat penting untuk model organisasi data InfluxDB. Untuk informasi selengkapnya, lihat dokumentasi [InfluxDB](#).

Menyebarkan dan memverifikasi aliran retensi

Setelah mengonfigurasi semua komponen aliran retensi data, Anda perlu menerapkan dan memverifikasi bahwa sistem berfungsi dengan benar. Verifikasi ini memastikan bahwa data industri Anda disimpan dengan benar secara lokal untuk akses dan analisis segera.

1. Hubungkan ketiga node seperti yang ditunjukkan pada diagram alir retensi data. Ini menciptakan pipeline lengkap dari langganan data ke penyimpanan lokal.



2. Pilih Deploy untuk menerapkan perubahan Anda dan mengaktifkan alur. Ini memulai proses pengumpulan dan penyimpanan data.
3. Gunakan InfluxDB Data Explorer untuk menanyakan dan memvisualisasikan data Anda. Alat ini memungkinkan Anda untuk memverifikasi bahwa data disimpan dengan benar dan untuk membuat visualisasi awal data deret waktu Anda.

Di Data Explorer, Anda harus dapat melihat pengukuran kecepatan angin Anda direkam dari waktu ke waktu, mengonfirmasi bahwa seluruh pipa dari pembuatan data ke penyimpanan lokal berfungsi dengan benar.

Untuk informasi selengkapnya, lihat [Kueri di Penjelajah Data](#) di InfluxData Documentation.

Dengan aliran publikasi data dan aliran retensi data yang diterapkan, Anda sekarang memiliki sistem lengkap yang mengirimkan data ke AWS IoT SiteWise cloud sambil mempertahankan salinan lokal untuk akses dan ketahanan langsung. Pendekatan jalur ganda ini memastikan bahwa Anda mendapatkan manfaat dari analitik dan penyimpanan berbasis cloud sambil mempertahankan visibilitas operasional di edge.

Siapkan Grafana untuk Edge SiteWise

Grafana® memungkinkan Anda membuat dasbor pemantauan real-time lokal untuk data industri Anda. Dengan memvisualisasikan data yang disimpan di InfluxDB®, Anda dapat memberikan wawasan langsung kepada operator tentang kinerja peralatan, efisiensi proses, dan potensi masalah. Visibilitas di tepi ini penting untuk operasi yang sensitif terhadap waktu dan menjaga kontinuitas selama gangguan jaringan.

Konfigurasi sumber data

Menghubungkan Grafana ke database InfluxDB Anda menciptakan lapisan visualisasi yang kuat untuk data industri Anda. Koneksi ini memungkinkan dasbor pemantauan waktu nyata yang dapat digunakan operator untuk membuat keputusan berdasarkan informasi tanpa dependensi cloud.

1. [Akses instans Grafana Anda secara lokal dengan menavigasi ke `http://127.0.0.1:3000` di browser Anda](#). Jika mengaktifkan TLS diperlukan, Anda dapat merujuk ke [Mengatur Grafana HTTPS untuk lalu lintas web yang aman](#) di Dokumentasi. Grafana Labs
2. Tambahkan sumber data InfluxDB yang menunjuk ke bucket seri waktu InfluxDB tempat Node-red menulis data. Misalnya, WindFarmData. Koneksi ini menetapkan tautan antara data yang Anda simpan dan platform visualisasi.
3. Untuk petunjuk terperinci, lihat [Mengkonfigurasi sumber data InfluxDB di Dokumentasi](#). Grafana Labs

Buat dasbor Grafana untuk SiteWise data Edge

Membuat dasbor adalah langkah terakhir dalam membangun solusi pemantauan lokal Anda. Dasbor memberikan representasi visual dari data industri Anda, sehingga lebih mudah untuk mengidentifikasi tren, anomali, dan potensi masalah secara sekilas.

- Ikuti panduan untuk membuat dasbor. Untuk informasi selengkapnya, lihat [Membangun dasbor pertama Anda](#) di Grafana LabsDokumentasi. Template ini mengasumsikan bucket Anda diberi nama WindFarmData dan pengukurannyaTurbineData.

Anda juga dapat menggunakan panduan mulai cepat dengan mengimpor contoh templat dasbor yang disediakan untuk membuat dasbor dengan cepat dengan plot deret waktu untuk data yang dihasilkan Node-red di bagian sebelumnya. Template ini menyediakan titik awal yang dapat Anda sesuaikan untuk memenuhi kebutuhan pemantauan spesifik Anda.

```
{
  "__inputs": [
    {
      "name": "DS_WINDFARM-DEMO",
      "label": "windfarm-demo",
      "description": "",
      "type": "datasource",
      "pluginId": "influxdb",
      "pluginName": "InfluxDB"
    }
  ],
  "__elements": {},
  "__requires": [
    {
      "type": "grafana",
      "id": "grafana",
      "name": "Grafana",
      "version": "11.6.0-pre"
    },
    {
      "type": "datasource",
      "id": "influxdb",
      "name": "InfluxDB",
      "version": "1.0.0"
    },
    {
      "type": "panel",
      "id": "timeseries",
      "name": "Time series",
      "version": ""
    }
  ],
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": {
```

```
        "type": "grafana",
        "uid": "-- Grafana --"
    },
    "enable": true,
    "hide": true,
    "iconColor": "rgba(0, 211, 255, 1)",
    "name": "Annotations & Alerts",
    "type": "dashboard"
}
]
},
"editable": true,
"fiscalYearStartMonth": 0,
"graphTooltip": 0,
"id": null,
"links": [],
"panels": [
    {
        "datasource": {
            "type": "influxdb",
            "uid": "${DS_WINDFARM-DEMO}"
        },
        "fieldConfig": {
            "defaults": {
                "color": {
                    "mode": "palette-classic"
                },
                "custom": {
                    "axisBorderShow": false,
                    "axisCenteredZero": false,
                    "axisColorMode": "text",
                    "axisLabel": "",
                    "axisPlacement": "auto",
                    "barAlignment": 0,
                    "barWidthFactor": 0.6,
                    "drawStyle": "line",
                    "fillOpacity": 0,
                    "gradientMode": "none",
                    "hideFrom": {
                        "legend": false,
                        "tooltip": false,
                        "viz": false
                    },
                    "insertNulls": false,
```

```
    "lineInterpolation": "linear",
    "lineWidth": 1,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "auto",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "mappings": [],
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {
        "color": "green"
      },
      {
        "color": "red",
        "value": 80
      }
    ]
  }
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
  "x": 0,
  "y": 0
},
"id": 1,
"options": {
  "legend": {
    "calcs": [],
    "displayMode": "list",
    "placement": "bottom",
```

```

        "showLegend": true
    },
    "tooltip": {
        "hideZeros": false,
        "mode": "single",
        "sort": "none"
    }
},
"pluginVersion": "11.6.0-pre",
"targets": [
    {
        "datasource": {
            "type": "influxdb",
            "uid": "${DS_WINDFARM-DEMO}"
        },
        "query": "from(bucket: \"WindFarmData\")\n  |> range(start:
v.timeRangeStart, stop: v.timeRangeStop)\n  |> filter(fn: (r) => r[\"_measurement
\"] == \"TurbineData\")\n  |> filter(fn: (r) => r[\"_field\"] == \"value\")\n
  |> filter(fn: (r) => r[\"name\"] == \"/Renton/WindFarm/Turbine/WindSpeed\")\n
  |> filter(fn: (r) => r[\"quality\"] == \"GOOD\")\n  |> aggregateWindow(every:
v.windowPeriod, fn: mean, createEmpty: false)\n  |> yield(name: \"mean\")",
        "refId": "A"
    }
],
"title": "Panel Title",
"type": "timeseries"
}
],
"schemaVersion": 41,
"tags": [],
"templating": {
    "list": []
},
"time": {
    "from": "now-6h",
    "to": "now"
},
"timepicker": {},
"timezone": "browser",
"title": "demo dashboard",
"uid": "fejc0t08o6d4wb",
"version": 1,
"weekStart": ""

```

```
}
```

Siapkan integrasi sumber terbuka dengan Docker (Linux)

Untuk proses penyebaran yang efisien, Anda dapat menggunakan Docker untuk menyiapkan Node-Red®, InfluxDB®, dan Grafana® di lingkungan Linux. Metode ini menggunakan kontainer yang telah dikonfigurasi sebelumnya, memungkinkan penyebaran cepat dan pengelolaan komponen yang lebih mudah.

Prasyarat pengaturan Docker

Sebelum Anda mulai, verifikasi yang memiliki yang berikut:

- Gateway V3 berkemampuan MQTT. Untuk informasi selengkapnya, lihat [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#).
- Plugin Docker Compose. Untuk langkah-langkah instalasi, lihat [Menginstal Docker Compose plugin di](#) dokumentasi Manual Docker.

Menyebarkan layanan

Penerapan ini menjalankan SiteWise Edge, InfluxDB, Node-red, dan Grafana pada host yang sama.

Mengatur lingkungan

1. Dapatkan akses root:

```
sudo -i
```

2. Buat file.env atau ekspor variabel lingkungan ini:

```
export INFLUXDB_PASSWORD=your-secure-influxdb-password
export INFLUXDB_TOKEN=your-secure-influxdb-token
export GRAFANA_PASSWORD=your-secure-grafana-password
```

Konfigurasikan jaringan Docker

- Buat jaringan jembatan menggunakan namaSiteWiseEdgeNodeRedDemoNetwork.

```
docker network create --driver=bridge SiteWiseEdgeNodeRedDemoNetwork
```

Perbarui penerapan SiteWise Edge

1. Arahkan ke [AWS IoT konsol](#)
2. Pilih perangkat Greengrass di menu navigasi kiri di bawah bagian Kelola, lalu Perangkat inti.
3. Pilih perangkat inti yang terhubung ke SiteWise Edge Gateway Anda.
4. Pilih tab Deployment, lalu pilih nilai ID Deployment.
5. Pilih Tindakan, lalu pilih Revisi.
6. Baca pesan pop up lalu pilih Revise Deployment.
7. Pada Langkah 2 - Pilih komponen, pilih komponen berikut dan kemudian pilih Berikutnya.
 - `aws.greengrass.clientdevices.mqtt.EMQX`
 - `aws.iot.SiteWiseEdgePublisher`
8. Pada Langkah 3 - Konfigurasi komponen, pilih nilai `aws.greengrass.clientdevices.mqtt.EMQX` komponen dan tambahkan konfigurasi jaringan berikut:

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      }
    },
    "authMode": "bypass",
    "dockerOptions": "-p 127.0.0.1:1883:1883 --network=SiteWiseEdgeNodeRedDemoNetwork",
    "requiresPrivilege": "true"
  }
}
```

9. Pilih Berikutnya.
10. Pada Langkah 4 - Konfigurasi pengaturan lanjutan, pilih Berikutnya.
11. Pilih Deploy

Luncurkan layanan

1. Mulai layanan menggunakan file Docker Compose. Jalankan perintah berikut di bawah direktori yang berisi `compose.yaml` file.

```
docker compose up -d
```

2. Buat terowongan SSH untuk mengakses layanan:

```
ssh -i path_to_your_ssh_key -L 1880:127.0.0.1:1880 -L 3000:127.0.0.1:3000 -L  
8086:127.0.0.1:8086 username@gateway_ip_address
```

Penerapan ini membuat layanan berikut di `SiteWiseEdgeNodeRedDemoNetwork` network:

InfluxDB v2 (port 8086)

Termasuk organisasi (`iot-sitewise-edge`), bucket `WindFarmData` InfluxDB, dan kredensial admin yang telah dikonfigurasi sebelumnya

Node-merah (port 1880)

Termasuk node InfluxDB dan alur pra-konfigurasi untuk integrasi AWS IoT SiteWise

Grafana (port 3000)

Termasuk pengguna admin, sumber data InfluxDB, dan dasbor pemantauan

Akses layanan

Setelah penerapan, akses layanan menggunakan yang berikut URLs dan kredensyal:

Note

Anda dapat mengakses setiap layanan dari host Anda atau mesin gateway.

Detail akses layanan

Layanan	URL	Kredensial
Simpul-merah	http://127.0.0.1:1880	Tidak ada kredensi yang diperlukan
InfluxDB	http://127.0.0.1:8086	Nama Pengguna: admin Kata sandi: \$ INFLUXDB_PASSWORD
Grafana	http://127.0.0.1:3000	Nama Pengguna: admin Kata sandi: \$GRAFANA_PASSWORD

Verifikasi penyebaran

Untuk memastikan penerapan Anda berhasil, lakukan pemeriksaan berikut:

1. Untuk Node-Red, verifikasi keberadaan dua aliran yang dimuat sebelumnya:
 - Aliran publikasi data
 - Aliran retensi data
2. Untuk AWS IoT SiteWise, di AWS IoT SiteWise konsol, konfirmasi keberadaan aliran data dengan alias/Renton/WindFarm/Turbine/WindSpeed.
3. Untuk InfluxDB, gunakan Data Explorer untuk memverifikasi penyimpanan data dalam TurbineData pengukuran di dalam bucket. WindFarmData
4. Untuk Grafana, lihat dasbor untuk mengonfirmasi tampilan data deret waktu yang dihasilkan dari Node-merah.

Memproses data untuk integrasi sumber terbuka

Data dapat diproses (seperti transformasi atau agregasi), pada tahap yang berbeda menggunakan berbagai alat, masing-masing melayani persyaratan pemantauan yang berbeda.

Memproses data dengan node Node-red

Ubah data Anda secara real time menggunakan node pemrosesan bawaan Node-Red®. Konfigurasi node ini melalui konsol Node-Red untuk membuat pipeline data Anda.

Node transformasi data

Ubah titik data individual, mirip dengan Transforms in AWS IoT SiteWise, menggunakan node ini:

- change node - Melakukan modifikasi nilai sederhana pada data Anda.
- fungsi node - Mengaktifkan JavaScript transformasi kustom untuk pemrosesan data yang kompleks.

Node perhitungan metrik

Gabungkan beberapa titik data menjadi satu output, mirip dengan Metrik di AWS IoT SiteWise, menggunakan node ini:

- batch node - Kelompokkan beberapa pesan untuk pemrosesan batch.
- join node - Menggabungkan beberapa aliran data menjadi satu output.
- aggregator node - Menghitung metrik agregat dari beberapa titik data.

Untuk opsi node tambahan, lihat Pustaka [Node-merah](#).

Buat tugas InfluxDB

Sementara Node-Red unggul dalam pemrosesan data dasar dengan pengaturan cepat, perhitungan metrik yang kompleks dapat menjadi tantangan dalam pemrograman berbasis aliran. InfluxDB® Tasks memberikan alternatif melalui skrip Flux terjadwal untuk kebutuhan pemrosesan lanjutan.

Gunakan Tugas InfluxDB untuk:

- Agregasi statistik di seluruh kumpulan data besar
- Operasi matematika pada beberapa properti
- Pengukuran turunan dari berbagai sumber

Fitur tugas

- Eksekusi Terjadwal - Jalankan tugas berdasarkan ekspresi cron

- Pemrosesan Batch - Optimalkan operasi untuk data deret waktu
- Pemulihan Kesalahan - Coba ulang operasi yang gagal secara otomatis
- Monitoring - Melacak eksekusi melalui log rinci

Kelola tugas melalui UI InfluxDB, API, atau CLI. Untuk informasi selengkapnya, lihat [Memproses data dengan tugas InfluxDB](#).

Gunakan transformasi Grafana

Ubah visualisasi data di Grafana® tanpa memodifikasi data sumber di InfluxDB. Transformasi Grafana hanya berlaku untuk lapisan visualisasi.

- Visual Builder - Buat transformasi tanpa menulis kode
- Live Preview - Lihat hasil transformasi secara real time
- Multi-Sumber - Memproses data dari berbagai sumber database
- Penyimpanan Efisien - Mengubah data pada waktu visualisasi tanpa menyimpan hasil perantara

Untuk informasi selengkapnya, lihat [Mengubah data](#).

Memecahkan masalah integrasi sumber terbuka

Untuk informasi selengkapnya tentang topik pemecahan masalah yang terkait dengan integrasi sumber terbuka untuk gateway SiteWise Edge, lihat [Memecahkan masalah integrasi sumber terbuka di Edge](#)

Aliran klasik, gateway V2 untuk Edge AWS IoT SiteWise

Pahami fitur dan batasan aliran Klasik, gateway V2 untuk Edge. AWS IoT SiteWise

Aliran Klasik, gateway V2 mempertahankan fungsionalitas tradisional yang akrab dari AWS IoT SiteWise penerapan sebelumnya sebelum pengenalan gateway V3 berkemampuan MQTT. Gerbang SiteWise Edge ini dianggap sebagai aliran Klasik, gateway V2. Mereka mempertahankan kompatibilitas mundur dan berfungsi dengan paket pemrosesan data. Sementara aliran Klasik, gateway V2 menawarkan kinerja yang andal untuk pengaturan yang ada, ia memiliki keterbatasan dibandingkan dengan opsi gateway yang lebih baru. Secara khusus, jenis gateway ini tidak sepenuhnya kompatibel dengan fitur-fitur canggih yang tersedia di tujuan gateway V3 berkemampuan MQTT. Untuk menggunakan protokol pesan MQTT, Anda dapat membuat gateway

V3 berkemampuan MQTT baru. Untuk informasi selengkapnya, lihat [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#).

Topik

- [Gunakan paket untuk mengumpulkan dan memproses data di SiteWise Edge](#)
- [Konfigurasi AWS IoT SiteWise komponen penerbit](#)
- [Tujuan dan pengelola AWS IoT Greengrass aliran](#)
- [Konfigurasi kemampuan tepi di AWS IoT SiteWise Edge](#)
- [Konfigurasi pemrosesan data tepi untuk AWS IoT SiteWise model dan aset](#)

Gunakan paket untuk mengumpulkan dan memproses data di SiteWise Edge

AWS IoT SiteWise Edge gateway menggunakan paket yang berbeda untuk menentukan cara mengumpulkan dan memproses data Anda.

Saat ini, paket berikut tersedia:

- Paket pengumpulan data — Gunakan paket ini untuk mengumpulkan data industri Anda dan mengarahkannya ke tujuan AWS Cloud. Secara default, paket ini diaktifkan secara otomatis untuk gateway SiteWise Edge Anda.
- Paket pemrosesan data — Gunakan paket ini untuk mengaktifkan komunikasi gateway SiteWise Edge dengan model dan aset aset yang dikonfigurasi tepi. Anda dapat menggunakan konfigurasi tepi untuk mengontrol data aset apa yang akan dihitung dan diproses di tempat. Anda kemudian dapat mengirim data Anda ke AWS IoT SiteWise atau AWS layanan lain. Untuk informasi selengkapnya tentang paket pemrosesan data, lihat [the section called “Konfigurasi pemrosesan data tepi”](#).

Paket upgrade

Important

Memutakhirkan versi paket pemrosesan data dari sebelumnya (dan termasuk) 2.0.x ke versi 2.1.x akan mengakibatkan hilangnya data pengukuran yang disimpan secara lokal.

SiteWise Edge gateway menggunakan paket yang berbeda untuk menentukan cara mengumpulkan dan memproses data Anda. Anda dapat menggunakan AWS IoT SiteWise konsol untuk meningkatkan paket.

Untuk meng-upgrade paket (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Dalam daftar Gateways, pilih gateway SiteWise Edge dengan paket yang ingin Anda tingkatkan.
4. Di bagian konfigurasi Gateway, pilih Pembaruan perangkat lunak tersedia.
5. Pada halaman Edit versi perangkat lunak, pilih Pembaruan.

 Note

Anda hanya dapat memutakhirkan paket yang diaktifkan. Untuk menemukan daftar paket yang diaktifkan untuk gateway SiteWise Edge ini, pilih Ikhtisar, lalu lihat bagian kemampuan Edge.

6. Pada halaman edit versi perangkat lunak, di bagian pembaruan komponen Gateway, lakukan hal berikut:
 - Untuk memperbarui kolektor OPC UA, pilih versi, lalu pilih Deploy.
 - Untuk memperbarui Publisher, pilih versi, lalu pilih Deploy.
 - Untuk memperbarui paket pemrosesan data, pilih versi, lalu pilih Deploy.
7. Setelah selesai menerapkan versi baru, pilih Selesai.

Jika Anda mengalami masalah dalam meningkatkan paket, lihat [Tidak dapat menerapkan paket ke gateway SiteWise Edge](#).

Konfigurasi AWS IoT SiteWise komponen penerbit

Setelah Anda membuat gateway AWS IoT SiteWise Edge dan menginstal perangkat lunak, Anda dapat mengatur komponen penerbit sehingga gateway SiteWise Edge Anda dapat mengekspor data ke AWS Cloud. Gunakan komponen penerbit untuk mengaktifkan fitur tambahan atau mengkonfigurasi pengaturan default. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise penerbit](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

 Note

Konfigurasi penerbit berbeda berdasarkan jenis gateway yang Anda gunakan. Untuk aliran Klasik, gateway V2, gunakan namespace. `iotsitewise:publisher:2` Untuk gateway V3 berkemampuan MQTT, gunakan namespace. `iotsitewise:publisher:3`

Console

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway SiteWise Edge yang ingin Anda konfigurasi penerbit.
4. Di bagian konfigurasi Publisher, pilih Edit
5. Untuk pesanan Penerbitan, pilih salah satu dari berikut ini:
 - Publikasikan data tertua terlebih dahulu — Gateway SiteWise Edge menerbitkan data tertua ke cloud terlebih dahulu secara default.
 - Publikasikan data terbaru terlebih dahulu — Gateway SiteWise Edge menerbitkan data terbaru ke cloud terlebih dahulu.
6. (Opsional) Jika Anda tidak ingin gateway SiteWise Edge mengompres data Anda, batalkan pilihan Aktifkan kompresi saat mengunggah data.
7. (Opsional) Jika Anda tidak ingin mempublikasikan data lama, pilih Kecualikan data kedaluwarsa dan lakukan hal berikut:
 - Untuk periode Cutoff, masukkan nilai dan pilih unit. Periode cutoff harus antara lima menit dan tujuh hari. Misalnya, jika batas waktu adalah tiga hari, data yang lebih tua dari tiga hari tidak dipublikasikan ke cloud.
8. (Opsional) Untuk mengatur pengaturan kustom tentang cara data ditangani pada perangkat lokal Anda, pilih Pengaturan penyimpanan lokal dan lakukan hal berikut:
 - a. Untuk periode Retensi, masukkan nomor dan pilih unit. Periode retensi harus antara satu menit dan 30 hari, dan lebih besar dari atau sama dengan periode rotasi. Misalnya, jika periode retensi adalah 14 hari, gateway SiteWise Edge menghapus data apa pun di tepi yang lebih lama dari periode batas yang ditentukan setelah disimpan selama 14 hari.
 - b. Untuk periode Rotasi, masukkan nomor dan pilih unit. Periode rotasi harus lebih besar dari satu menit, dan sama dengan, atau kurang dari, periode retensi. Misalnya,

katakanlah periode rotasi adalah dua hari, gateway SiteWise Edge mengumpulkan dan menyimpan data yang lebih lama dari periode cutoff ke satu file. Untuk gateway yang dihosting sendiri AWS IoT Greengrass V2, gateway SiteWise Edge mentransfer sekumpulan data ke direktori lokal berikut setiap dua hari sekali: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`

- c. Untuk kapasitas Penyimpanan, masukkan nilai yang lebih besar dari atau sama dengan 1. Jika kapasitas penyimpanan 2 GB, gateway SiteWise Edge mulai menghapus data ketika lebih dari 2 GB data disimpan secara lokal.

9. Pilih Simpan.

AWS CLI

Gunakan [UpdateGatewayCapabilityConfiguration](#) API untuk mengonfigurasi penerbit.

Atur parameter `capabilityNamespace` ke `iotsitewise:publisher:2`.

Example : Konfigurasi penerbit untuk Classic Stream, gateway V2

Namespace penerbit: `iotsitewise:publisher:2`

```
{
  "SiteWisePublisherConfiguration": {
    "publishingOrder": "TIME_ORDER",
    "enableCompression": true,
    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportSizeLimitGB": 10
      }
    }
  },
  "SiteWiseS3PublisherConfiguration": {
    "accessRoleArn": "arn:aws:iam:123456789012:role/roleName",
    "streamToS3ConfigMapping": [
      {
        "streamName": "S3_OPC-UA_Data_Collector",
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/dataCollector",
        "publishPolicy": {
          "publishFrequency": "10m",
```



```

        "localSizeLimitGB": 10
      },
      "siteWiseImportPolicy": {
        "enableSiteWiseStorageImport": true,
        "enableDeleteAfterImport": true
      }
    ]
  }
}

```

Publisher menyediakan parameter konfigurasi berikut yang dapat Anda sesuaikan:

SiteWisePublisherConfiguration

publishingOrder

Urutan di mana data dipublikasikan ke cloud. Nilai parameter ini dapat berupa salah satu dari yang berikut:

- **TIME_ORDER**(Publikasikan data tertua terlebih dahulu) — Data paling awal dipublikasikan ke cloud terlebih dahulu, secara default.
- **RECENT_DATA**(Publikasikan data terbaru terlebih dahulu) — Data terbaru dipublikasikan ke cloud terlebih dahulu.

enableCompression

Setel ini untuk **true** mengompres data sebelum dipublikasikan. Kompresi data dapat mengurangi penggunaan bandwidth.

dropPolicy

(Opsional) Kebijakan yang mengontrol data apa yang dipublikasikan ke cloud.

cutoffAge

Usia maksimum data yang akan dipublikasikan ditentukan dalam hari, jam, dan menit. Misalnya, 7d atau 1d7h16m. Data yang lebih tua dari yang Anda tentukan tidak dikirim ke AWS IoT SiteWise.

Data yang lebih awal dari periode cutoff tidak dipublikasikan ke cloud. Usia batas harus antara lima menit dan tujuh hari.

Anda dapat menggunakan **m**, **h**, dan **d** ketika Anda menentukan batas usia. Catatan yang **m** mewakili menit, **h** mewakili jam, dan **d** mewakili hari.

`exportPolicy`

(Opsional) Kebijakan yang mengelola penyimpanan data di tepi. Kebijakan ini berlaku untuk data yang lebih awal dari batas usia.

`retentionPeriod`

Gateway SiteWise Edge Anda menghapus data apa pun di tepi yang lebih awal dari periode cutoff dari penyimpanan lokal setelah disimpan untuk periode retensi yang ditentukan. Periode retensi harus antara satu menit dan 30 hari, dan lebih besar dari atau sama dengan periode rotasi.

Anda dapat menggunakan `m`, `h`, dan `d` ketika Anda menentukan periode retensi. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari.

`rotationPeriod`

Interval waktu untuk mengumpulkan dan menyimpan data yang lebih awal dari periode cutoff ke satu file. Gateway SiteWise Edge mentransfer satu batch data ke direktori lokal berikut pada akhir setiap periode rotasi: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`. Periode rotasi harus lebih besar dari satu menit, dan sama dengan atau kurang dari periode retensi.

Anda dapat menggunakan `m`, `h`, dan `d` ketika Anda menentukan periode rotasi. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari.

`exportSizeLimitGB`

Ukuran maksimum yang diizinkan dari data yang disimpan secara lokal, dalam GB. Jika kuota ini dilanggar, gateway SiteWise Edge mulai menghapus data paling awal hingga ukuran data yang disimpan secara lokal sama dengan atau kurang dari kuota. Nilai parameter ini harus lebih besar dari atau sama dengan 1.

`SiteWiseS3PublisherConfiguration`

`accessRoleArn`

Peran akses yang memberikan AWS IoT SiteWise izin untuk mengelola bucket Amazon S3 yang Anda terbitkan.

`streamToS3ConfigMapping`

Array konfigurasi yang memetakan aliran ke konfigurasi Amazon S3.

`streamName`

Aliran untuk membaca dan mempublikasikan ke konfigurasi Amazon S3.

`targetBucketArn`

Ember ARN untuk dipublikasikan ke.

`publishPolicy`

`publishFrequency`

Frekuensi penerbitan gateway SiteWise Edge ke bucket Amazon S3.

`localSizeLimitGB`

Ukuran maksimum file yang ditulis ke disk lokal. Jika ambang batas ini dilanggar, penerbit menerbitkan semua data buffer ke tujuannya.

`siteWiseImportPolicy`

`enableSiteWiseStorageImport`

Setel ini `true` untuk mengimpor data dari bucket Amazon S3 ke AWS IoT SiteWise penyimpanan.

`enableDeleteAfterImport`

Setel ini `true` untuk menghapus file di bucket Amazon S3 setelah tertelan ke penyimpanan. AWS IoT SiteWise

Tujuan dan pengelola AWS IoT Greengrass aliran

AWS IoT Greengrass stream manager memungkinkan Anda mengirim data ke AWS Cloud tujuan berikut: saluran masuk AWS IoT Analytics, streaming di Amazon Kinesis Data Streams, properti aset di, atau objek AWS IoT SiteWise di Amazon Simple Storage Service (Amazon S3). Untuk informasi selengkapnya, lihat [Mengelola aliran data di Panduan AWS IoT Greengrass Version 2 Pengembang AWS IoT Greengrass Inti](#).

Example : Struktur pesan aliran data

Contoh berikut menunjukkan struktur pesan aliran data yang diperlukan dikirimkan oleh manajer AWS IoT Greengrass aliran.

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

Note

Pesan aliran data harus menyertakan (`assetId` dan `propertyId`) atau `propertyAlias` dalam strukturnya.

`assetId`

(Opsional) ID aset yang akan diperbarui.

`propertyAlias`

(Opsional) Alias yang mengidentifikasi properti, seperti jalur aliran data server OPC UA. Misalnya:

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

Untuk informasi selengkapnya, lihat [Mengelola aliran data](#) di Panduan AWS IoT SiteWise Pengguna.

`propertyId`

(Opsional) ID properti aset untuk entri ini.

propertyValues

(Wajib) Daftar nilai properti yang akan diunggah. Anda dapat menentukan hingga 10 elemen `propertyValues` array.

quality

(Opsional) Kualitas nilai properti aset.

timestamp

(Wajib) Stempel waktu dari nilai properti aset.

offsetInNanos

(Opsional) Offset nanodetik dari. `timeInSeconds`

timeInSeconds

(Wajib) Tanggal stempel waktu, dalam hitungan detik, dalam format epoch Unix. Data nanodetik pecahan disediakan oleh. `offsetInNanos`

value

(Wajib) Nilai properti aset.

Note

Hanya satu dari nilai berikut yang dapat ada di `value` lapangan.

booleanValue

(Opsional) Data properti aset tipe Boolean (`true` atau `false`).

doubleValue

(Opsional) Data properti aset tipe ganda (nomor floating point).

integerValue

(Opsional) Data properti aset tipe integer (bilangan bulat).

stringValue

(Opsional) Data properti aset tipe string (urutan karakter).

Konfigurasi kemampuan tepi di AWS IoT SiteWise Edge

Anda dapat menggunakan AWS IoT SiteWise Edge untuk mengumpulkan dan menyimpan sementara data sehingga Anda dapat mengatur dan memproses data perangkat secara lokal. Dengan mengaktifkan pemrosesan tepi, Anda dapat memilih untuk mengirim hanya data agregat ke AWS Cloud untuk mengoptimalkan penggunaan bandwidth dan biaya penyimpanan cloud Anda. Menggunakan AWS IoT SiteWise komponen dengan AWS IoT Greengrass, Anda dapat mengumpulkan dan memproses data di tepi sebelum mengirimnya ke AWS Cloud, atau mengelolanya di tempat menggunakan SiteWise Edge APIs.

Pengumpulan data terjadi melalui paket data dan AWS IoT SiteWise komponen yang berjalan AWS IoT Greengrass.

Note

- AWS IoT SiteWise mempertahankan data tepi Anda di gateway SiteWise Edge Anda hingga 30 hari. Periode penyimpanan data Anda tergantung pada ruang disk yang tersedia di perangkat Anda.
- Jika gateway SiteWise Edge Anda telah terputus AWS Cloud selama 30 hari, [Paket Pemrosesan Data](#) secara otomatis dinonaktifkan.

Topik

- [Siapkan kemampuan tepi di SiteWise Edge](#)

Siapkan kemampuan tepi di SiteWise Edge

AWS IoT SiteWise menyediakan paket berikut yang dapat digunakan gateway SiteWise Edge Anda untuk menentukan cara mengumpulkan dan memproses data Anda. Pilih paket untuk mengaktifkan kemampuan tepi untuk gateway SiteWise Edge Anda.

- Paket pengumpulan data memungkinkan gateway SiteWise Edge Anda mengumpulkan data dari beberapa server OPC UA, lalu mengeksport data dari edge ke AWS Cloud. Ini menjadi aktif setelah Anda menambahkan sumber data ke gateway SiteWise Edge Anda.
- Paket pemrosesan data memungkinkan gateway SiteWise Edge Anda untuk memproses data peralatan Anda di tepi. Misalnya, Anda dapat menggunakan model aset untuk menghitung metrik

dan transformasi. Untuk informasi selengkapnya tentang model aset dan aset, lihat [Model aset industri](#).

 Note

- Paket pemrosesan data hanya tersedia di platform x86.

Untuk mengkonfigurasi kemampuan tepi

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway SiteWise Edge yang ingin Anda aktifkan kemampuan tepi.
4. Di bagian kemampuan Edge, pilih Edit
5. Di bagian kemampuan Edge, pilih Aktifkan paket pemrosesan data (dikenakan biaya tambahan).
6. (Opsional) Di bagian koneksi Edge LDAP, Anda dapat memberikan grup pengguna di direktori perusahaan Anda akses ke gateway SiteWise Edge ini. Grup pengguna dapat menggunakan kredensi Lightweight Directory Access Protocol (LDAP) untuk mengakses gateway Edge. SiteWise Kemudian mereka dapat menggunakan AWS IoT SiteWise aplikasi AWS OpsHub for, operasi AWS IoT SiteWise API, atau alat lain untuk mengelola gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Kelola SiteWise gateway Edge](#).

 Note

Anda juga dapat menggunakan Linux atau Microsoft Windows kredensial untuk mengakses gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Akses gateway SiteWise Edge Anda menggunakan kredensial sistem operasi Linux](#).

- a. Pilih Diaktifkan.
- b. Untuk nama Penyedia, masukkan nama untuk penyedia LDAP Anda.
- c. Untuk nama host atau alamat IP, masukkan nama host atau alamat IP server LDAP Anda.
- d. Untuk Port, masukkan nomor port.
- e. Untuk Base Distinguished Name (DN), masukkan Distinguished Name (DN) untuk basis.

Jenis atribut berikut didukung: commonName (CN), localityName (L), Name (ST), OrganizationName (O), stateOrProvince (OU), countryName (C), streetAddress organizationalUnitName (STREET), domainComponent (DC), dan userid (UID).

- f. Untuk grup Admin DN, masukkan DN.
- g. Untuk grup Pengguna DN, masukkan DN.

7. Pilih Simpan.

Sekarang setelah Anda mengaktifkan kemampuan edge di gateway SiteWise Edge Anda, Anda perlu mengonfigurasi model aset Anda untuk edge. Konfigurasi tepi model aset Anda menentukan tempat properti aset Anda dihitung. Anda dapat menghitung semua properti di edge, atau Anda dapat mengonfigurasi properti model aset Anda secara terpisah. [Properti model aset meliputi metrik, transformasi, dan pengukuran.](#)

Untuk informasi selengkapnya tentang properti aset, lihat [the section called “Tentukan properti data”](#).

Setelah Anda membuat model aset Anda, Anda kemudian dapat mengonfigurasinya untuk edge. Untuk informasi selengkapnya tentang mengonfigurasi model aset Anda untuk edge, lihat [the section called “Buat model aset \(konsol\)”](#).

Note

Model aset dan dasbor secara otomatis disinkronkan antara AWS Cloud dan gateway SiteWise Edge Anda setiap 10 menit. Anda juga dapat menyinkronkan secara manual dari aplikasi gateway SiteWise Edge lokal.

Konfigurasikan pemrosesan data tepi untuk AWS IoT SiteWise model dan aset

Anda dapat menggunakan AWS IoT SiteWise Edge untuk mengumpulkan, menyimpan, mengatur, dan memantau data peralatan secara lokal. Anda dapat menggunakan SiteWise Edge sehingga Anda dapat memodelkan data industri Anda dan SiteWise Monitor untuk membuat dasbor bagi staf operasional Anda untuk memvisualisasikan data secara lokal. Anda dapat memproses data secara lokal dan mengirimkannya ke AWS Cloud, atau memprosesnya secara lokal menggunakan API. AWS IoT SiteWise

Dengan AWS IoT SiteWise Edge, Anda dapat memproses data mentah secara lokal dan memilih untuk mengirim hanya data gabungan ke AWS Cloud untuk mengoptimalkan penggunaan bandwidth dan biaya penyimpanan cloud Anda.

 Note

- AWS IoT SiteWise mempertahankan data tepi Anda di gateway SiteWise Edge Anda hingga 30 hari. Periode penyimpanan data Anda tergantung pada ruang disk yang tersedia di perangkat Anda.
- Jika gateway SiteWise Edge Anda telah terputus dari AWS Cloud selama 30 hari, gateway Edge akan dinonaktifkan [Siapkan sumber OPC UA di SiteWise Edge](#) secara otomatis.

Konfigurasi model aset untuk pemrosesan data di SiteWise Edge

Anda harus mengonfigurasi model aset Anda untuk edge sebelum Anda dapat memproses data gateway SiteWise Edge Anda di tepi. Konfigurasi tepi model aset Anda menentukan tempat properti aset Anda dihitung. Anda dapat memilih untuk menghitung semua properti di edge dan mengirim hasilnya ke AWS Cloud, atau menyesuaikan tempat menghitung setiap properti aset secara terpisah. Untuk informasi selengkapnya, lihat [Konfigurasi pemrosesan data tepi untuk AWS IoT SiteWise model dan aset](#).

Properti aset meliputi metrik, transformasi, dan pengukuran:

- Metrik adalah data agregat aset selama periode waktu tertentu. Anda dapat menghitung metrik baru dengan menggunakan data metrik yang ada. AWS IoT SiteWise selalu mengirimkan metrik Anda ke AWS Cloud untuk penyimpanan jangka panjang. AWS IoT SiteWise menghitung metrik di AWS Cloud secara default. Anda dapat mengonfigurasi model aset Anda untuk menghitung metrik Anda di tepi. AWS IoT SiteWise mengirimkan hasil yang diproses ke AWS Cloud.
- Transformasi adalah ekspresi matematika yang memetakan titik data properti aset dari satu bentuk ke bentuk lainnya. Transformasi dapat menggunakan metrik sebagai data input dan harus dihitung dan disimpan di lokasi yang sama dengan inputnya. Jika Anda mengonfigurasi input metrik untuk menghitung di tepi, hitung AWS IoT SiteWise juga transformasi terkaitnya di tepi.
- Pengukuran diformat sebagai data mentah yang dikumpulkan dan dikirim perangkat Anda ke AWS Cloud secara default. Anda dapat mengonfigurasi model aset untuk menyimpan data ini di perangkat lokal Anda.

Untuk informasi selengkapnya tentang properti aset, lihat [the section called “Tentukan properti data”](#).

Setelah Anda membuat model aset Anda, Anda kemudian dapat mengonfigurasinya untuk edge. Untuk informasi selengkapnya tentang mengonfigurasi model aset Anda untuk edge, lihat [the section called “Buat model aset \(konsol\)”](#).

 Note

Model aset dan dasbor secara otomatis disinkronkan antara AWS Cloud dan gateway SiteWise Edge Anda setiap 10 menit. Anda juga dapat menyinkronkan secara manual dari file [Kelola SiteWise gateway Edge](#).

Anda dapat menggunakan AWS IoT SiteWise REST APIs dan AWS Command Line Interface (AWS CLI) untuk menanyakan gateway SiteWise Edge Anda untuk data di tepi. Sebelum Anda menanyakan gateway SiteWise Edge Anda untuk data di tepi, Anda harus memenuhi prasyarat berikut:

- Kredensial Anda harus ditetapkan untuk REST. APIs Untuk informasi selengkapnya tentang menyetel kredensial, lihat [the section called “Kelola gateway”](#)
- Titik akhir SDK harus mengarah ke alamat IP gateway SiteWise Edge Anda. Anda dapat menemukan informasi selengkapnya di dokumentasi SDK Anda. Misalnya, lihat [Menentukan Titik Akhir Kustom di Panduan AWS SDK for Java 2.x](#) Pengembang.
- Sertifikat gateway SiteWise Edge Anda harus terdaftar. Anda dapat menemukan informasi selengkapnya tentang mendaftarkan sertifikat gateway SiteWise Edge Anda di dokumentasi untuk SDK Anda. Misalnya, lihat [Mendaftarkan Bundel Sertifikat di Node.js](#) di Panduan AWS SDK for Java 2.x Pengembang.

Untuk informasi selengkapnya tentang kueri data dengan AWS IoT SiteWise, lihat [Data kueri dari AWS IoT SiteWise](#).

Tambahkan sumber data ke gateway AWS IoT SiteWise Edge Anda

Setelah menyiapkan gateway AWS IoT SiteWise Edge, Anda dapat menambahkan dan mengonfigurasi sumber data untuk menyerap data dari peralatan industri lokal. AWS IoT SiteWise SiteWise Edge mendukung berbagai protokol, termasuk OPC UA, dan banyak protokol lain yang tersedia melalui sumber data mitra. Sumber-sumber ini memungkinkan gateway Anda untuk

terhubung dengan server lokal dan mengambil data industri Anda. Dengan mengonfigurasi sumber data, Anda dapat mengambil data dari berbagai sumber data, dan kemudian mengaitkan aliran data dengan properti aset, memungkinkan pemodelan aset industri yang komprehensif dan pemetaan data. AWS IoT SiteWise

Topik

- [Sumber data OPC UA untuk gateway AWS IoT SiteWise Edge](#)
- [Sumber data mitra di gateway SiteWise Edge](#)

Sumber data OPC UA untuk gateway AWS IoT SiteWise Edge

Setelah menyiapkan gateway AWS IoT SiteWise Edge, Anda dapat mengonfigurasi sumber data sehingga gateway SiteWise Edge Anda dapat menyerap data dari peralatan industri lokal ke AWS IoT SiteWise. Setiap sumber mewakili server lokal, seperti server OPC UA, yang menghubungkan gateway SiteWise Edge Anda dan mengambil aliran data industri. Untuk informasi selengkapnya tentang menyiapkan gateway SiteWise Edge, lihat [Buat gateway SiteWise Edge yang dihosting sendiri](#).

Jenis gateway, berkemampuan MQTT, gateway V3 versus aliran Klasik, gateway V2, memengaruhi cara data OPC UA ditangani. Dalam aliran Klasik, gateway V2, sumber data OPC UA ditambahkan langsung ke konfigurasi penerbit IoT gateway. SiteWise Setiap sumber data digabungkan dengan gateway, dan perutean data dikonfigurasi secara individual untuk setiap sumber. Sebaliknya, menggunakan gateway V3 berkemampuan MQTT, sumber data OPC UA dikonversi ke topik MQTT dan dikelola melalui tujuan terpusat. Untuk informasi lebih lanjut tentang setiap jenis, lihat [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#) dan [Aliran klasik, gateway V2 untuk Edge AWS IoT SiteWise](#).

Note

AWS IoT SiteWise restart gateway SiteWise Edge Anda setiap kali Anda menambahkan atau mengedit sumber. Gateway SiteWise Edge Anda tidak akan menelan data saat memperbarui konfigurasi sumber. Waktu untuk memulai ulang gateway SiteWise Edge Anda tergantung pada jumlah tag pada sumber gateway SiteWise Edge Anda. Waktu restart dapat berkisar dari beberapa detik (untuk gateway SiteWise Edge dengan beberapa tag) hingga beberapa menit (untuk gateway SiteWise Edge dengan banyak tag).

Setelah membuat sumber, Anda dapat mengaitkan aliran data Anda dengan properti aset. Untuk informasi selengkapnya tentang cara membuat dan menggunakan aset, lihat [Model aset industri](#).

Anda dapat melihat CloudWatch metrik untuk memverifikasi bahwa sumber data terhubung AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [AWS IoT Greengrass Version 2 metrik gateway](#).

Saat ini, AWS IoT SiteWise mendukung protokol sumber data berikut:

- [OPC UA](#) — Protokol komunikasi machine-to-machine (M2M) untuk otomasi industri.

Support untuk protokol industri tambahan

SiteWise Edge mendukung berbagai protokol industri melalui integrasi dengan mitra sumber data. Kemitraan ini memungkinkan konektivitas dengan lebih dari 200 protokol yang berbeda, mengakomodasi berbagai sistem dan perangkat industri.

Untuk daftar mitra sumber data yang tersedia, lihat [SiteWise Opsi sumber data mitra gateway tepi](#).

Siapkan sumber OPC UA di SiteWise Edge

Anda dapat menggunakan AWS IoT SiteWise konsol atau kemampuan gateway SiteWise Edge untuk menentukan dan menambahkan sumber OPC UA ke gateway SiteWise Edge Anda untuk mewakili server OPC UA lokal.

Topik

- [Konfigurasi sumber OPC UA \(konsol\)](#)
- [Konfigurasi sumber OPC UA \(\)AWS CLI](#)

Konfigurasi sumber OPC UA (konsol)

Anda dapat menggunakan konsol untuk mengonfigurasi sumber OPC UA dengan prosedur berikut.

 Note

Peringatan: Duplikat TQVs dapat mengakibatkan pengisian ganda.

Untuk mengonfigurasi sumber OPC UA menggunakan konsol AWS IoT SiteWise

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway SiteWise Edge untuk menambahkan sumber OPC UA.
4. Pilih Tambahkan sumber data.
5. Masukkan nama untuk sumbernya.
6. Masukkan titik akhir lokal dari server sumber data. Titik akhir dapat berupa alamat IP atau nama host. Anda juga dapat menambahkan nomor port ke titik akhir lokal. Misalnya, titik akhir lokal Anda mungkin terlihat seperti ini: **opc.tcp://203.0.113.0:49320**
7. (Opsional) Untuk ID Node untuk pemilihan, tambahkan filter node untuk membatasi aliran data mana yang tertelan ke Cloud. AWS Secara default, gateway SiteWise Edge menggunakan simpul akar server untuk menelan semua aliran data. Anda dapat menggunakan filter node untuk mengurangi waktu startup dan penggunaan CPU gateway SiteWise Edge Anda dengan hanya menyertakan jalur ke data yang Anda modelkan AWS IoT SiteWise. Secara default, gateway SiteWise Edge mengunggah semua jalur OPC UA kecuali yang dimulai dengan. /Server/ Untuk menentukan filter node OPC UA, Anda dapat menggunakan jalur node * dan ** karakter wildcard. Untuk informasi selengkapnya, lihat [Gunakan filter node OPC UA di SiteWise Edge](#).
8. Tujuan bervariasi antara gateway V3 berkemampuan MQTT, dan aliran Klasik, gateway V2.
 - Steam klasik, tujuan gateway V2 memiliki hubungan 1:1 dengan sumbernya. Setiap sumber mengirimkan data ke tujuan tertentu.
 - Tujuan gateway V3 berkemampuan MQTT disiapkan secara terpisah karena model hub dan spoke memungkinkan Anda memusatkan konfigurasi dan pengelolaan beberapa sumber data di seluruh gateway yang berbeda. Untuk mengatur tujuan di gateway V3, lihat [Memahami tujuan AWS IoT SiteWise Edge](#).

Classic steams, V2 gateway destinations

- AWS IoT SiteWise real-time — Pilih ini untuk mengirim data langsung ke AWS IoT SiteWise penyimpanan. Menelan dan memantau data secara real-time di tepi.
- AWS IoT SiteWise Buffer menggunakan Amazon S3 - Kirim data dalam format Parquet ke Amazon S3 dan kemudian impor ke penyimpanan. AWS IoT SiteWise Pilih opsi ini untuk menelan data dalam batch, dan menyimpan data historis dengan cara yang hemat biaya. Anda dapat mengonfigurasi lokasi bucket Amazon S3 pilihan Anda, dan frekuensi di mana Anda ingin data diunggah ke Amazon S3. Anda juga dapat memilih apa yang harus dilakukan dengan data setelah tertelan. AWS IoT SiteWise Anda dapat memilih untuk memiliki data yang tersedia di keduanya AWS IoT SiteWise dan Amazon S3 atau Anda

dapat memilih untuk menghapusnya secara otomatis dari Amazon S3 setelah diimpor ke AWS IoT SiteWise

- Bucket Amazon S3 adalah mekanisme pementasan dan buffering dan mendukung file dalam format Parquet.
- Jika Anda memilih kotak centang Impor data ke AWS IoT SiteWise penyimpanan, data diunggah ke Amazon S3 terlebih dahulu, lalu AWS IoT SiteWise ke penyimpanan.
 - Jika Anda memilih kotak centang Hapus data dari Amazon S3, data dihapus dari Amazon S3, setelah diimpor ke penyimpanan. SiteWise
 - Jika Anda menghapus kotak centang Hapus data dari Amazon S3, data disimpan baik di Amazon S3, dan di penyimpanan. SiteWise
- Jika Anda menghapus kotak centang Impor data ke AWS IoT SiteWise penyimpanan, data hanya disimpan di Amazon S3. Itu tidak diimpor ke SiteWise penyimpanan.

Kunjungi [Kelola penyimpanan data](#) untuk detail tentang berbagai opsi penyimpanan yang AWS IoT SiteWise disediakan. Untuk mempelajari lebih lanjut tentang opsi harga, lihat [AWS IoT SiteWise harga](#).

- AWS IoT Greengrass pengelola aliran — Gunakan pengelola AWS IoT Greengrass aliran untuk mengirim data ke tujuan AWS Cloud berikut: saluran di AWS IoT Analytics, streaming di Amazon Kinesis Data Streams, properti aset di, atau objek AWS IoT SiteWise di Amazon Simple Storage Service (Amazon S3). Untuk informasi selengkapnya, lihat [Mengelola aliran data di Panduan AWS IoT Greengrass Version 2 Pengembang AWS IoT Greengrass](#) Inti.

Masukkan nama untuk AWS IoT Greengrass streaming.

MQTT-enabled, V3 gateway destinations

1. Lihat [Berkemampuan MQTT, Gateway V3 untuk Edge AWS IoT SiteWise](#) untuk informasi tentang menambahkan tujuan Anda yang relevan.
 2. Kembali ke prosedur ini setelah menambahkan tujuan sumber Anda.
9. Di panel konfigurasi lanjutan, Anda dapat melakukan hal berikut:
- a. Pilih mode keamanan Pesan untuk koneksi dan data dalam perjalanan antara server sumber dan gateway SiteWise Edge Anda. Bidang ini adalah kombinasi dari kebijakan keamanan OPC UA dan mode keamanan pesan. Pilih kebijakan keamanan dan mode keamanan pesan yang sama yang Anda tentukan untuk server OPC UA Anda.

- b. Jika sumber Anda memerlukan otentikasi, pilih AWS Secrets Manager rahasia dari daftar konfigurasi Otentikasi. Gateway SiteWise Edge menggunakan kredensial otentikasi dalam rahasia ini ketika terhubung ke sumber data ini. Anda harus melampirkan rahasia ke AWS IoT Greengrass komponen gateway SiteWise Edge Anda untuk menggunakannya untuk otentikasi sumber data. Untuk informasi selengkapnya, lihat [the section called “Konfigurasi otentikasi sumber data”](#).

 Tip

Server data Anda mungkin memiliki opsi bernama Izinkan login anonim. Jika opsi ini adalah Ya, maka sumber Anda tidak memerlukan otentikasi.

- c. (Opsional) Anda dapat mengaktifkan awalan aliran data dengan memilih Aktifkan awalan aliran data - opsional.
- Masukkan awalan aliran Data. Gateway SiteWise Edge menambahkan awalan ini ke semua aliran data dari sumber ini. Gunakan awalan aliran data untuk membedakan antara aliran data yang memiliki nama yang sama dari sumber yang berbeda. Setiap aliran data harus memiliki nama unik dalam akun Anda.
- d. (Opsional) Pilih opsi konversi tipe data untuk mengonversi tipe data OPC UA yang tidak didukung menjadi string sebelum memasukkannya ke dalam. AWS IoT SiteWise Konversi nilai array dengan tipe data sederhana ke string JSON dan tipe DateTime data ke string ISO 8601. Untuk informasi selengkapnya, lihat [Mengonversi tipe data yang tidak didukung](#).
- e. (Opsional) Untuk grup Properti, pilih Tambahkan grup baru.
- i. Masukkan Nama untuk grup properti.
 - ii. Untuk Properti:
 1. Untuk jalur Node, tambahkan filter node OPC UA untuk membatasi jalur OPC UA mana yang diunggah. AWS IoT SiteWise Formatnya mirip dengan Node ID untuk seleksi.
 - iii. Untuk pengaturan Grup, lakukan hal berikut:
 1. Untuk pengaturan kualitas Data, pilih jenis kualitas data yang ingin Anda konsumsi oleh AWS IoT SiteWise Kolektor.
 2. Untuk pengaturan mode Pindai, konfigurasi properti langganan standar menggunakan mode Pindai. Anda dapat memilih Subscribe atau Poll. Untuk

informasi selengkapnya tentang mode pemindaian, lihat [the section called “Filter rentang konsumsi data”](#).

Subscribe

Untuk mengirim setiap titik data

- i. Pilih Berlangganan dan atur yang berikut:
 - A. [Pemicu perubahan data](#) — Kondisi yang memulai peringatan perubahan data.
 - B. [Ukuran antrian langganan](#) — Kedalaman antrian pada server OPC-UA untuk metrik tertentu di mana pemberitahuan untuk item yang dipantau diantrian.
 - C. [Interval penerbitan langganan](#) — Interval (dalam milidetik) siklus penerbitan yang ditentukan saat langganan dibuat.
 - D. Interval snapshot - Opsional - Pengaturan batas waktu frekuensi snapshot untuk memastikan bahwa AWS IoT SiteWise Edge menyerap aliran data yang stabil.
 - E. Tingkat pemindaian — Tingkat yang Anda inginkan gateway SiteWise Edge untuk membaca register Anda. AWS IoT SiteWise secara otomatis menghitung tingkat pemindaian minimum yang diijinkan untuk gateway SiteWise Edge Anda.
 - F. Timestamp — Stempel waktu untuk disertakan dengan titik data OPC UA Anda. Anda dapat menggunakan stempel waktu server atau stempel waktu perangkat Anda.



Note

Gunakan versi 2.5.0 atau yang lebih baru dari komponen kolektor SiteWise IoT OPC UA. Jika Anda menggunakan fitur stempel waktu dengan versi sebelumnya, pembaruan konfigurasi gagal. Untuk informasi selengkapnya, lihat [Perbarui versi AWS IoT SiteWise komponen](#).

- ii. Dalam pengaturan Deadband, konfigurasi tipe Deadband. Jenis deadband mengontrol data apa yang dikirim sumber Anda ke Anda AWS IoT

SiteWise, dan data apa yang dibuang. Untuk informasi lebih lanjut tentang pengaturan deadband, lihat [the section called “Filter rentang konsumsi data”](#).

- Tidak ada — Server terkait mengirimkan semua titik data untuk grup properti ini.
 - Persentase — Server terkait hanya mengirimkan data yang berada di luar persentase tertentu dari rentang data. Rentang ini dihitung oleh server berdasarkan unit teknik minimum dan maksimum yang ditentukan untuk setiap node. Jika server tidak mendukung persentase deadband atau tidak memiliki unit teknik yang ditentukan, gateway menghitung rentang menggunakan nilai minimum dan maksimum yang disediakan di bawah ini.
 - Absolute — Server terkait hanya mengirimkan data yang berada di luar rentang tertentu.
- A. Tetapkan nilai Deadband sebagai persentase rentang data ke deadband.
 - B. (Opsional) Tentukan minimum dan maksimum untuk rentang deadband menggunakan Rentang minimum - opsional dan Rentang maksimum - opsional.

Poll

Untuk mengirim titik data pada interval tertentu

- Pilih Poll dan atur yang berikut:
 - A. Tingkat pemindaian — Tingkat yang Anda inginkan gateway SiteWise Edge untuk membaca register Anda. AWS IoT SiteWise secara otomatis menghitung tingkat pemindaian minimum yang diijinkan untuk gateway SiteWise Edge Anda.
 - B. Timestamp — Stempel waktu untuk disertakan dengan titik data OPC UA Anda. Anda dapat menggunakan stempel waktu server atau stempel waktu perangkat Anda.



Note

Gunakan versi 2.5.0 atau yang lebih baru dari komponen kolektor SiteWise IoT OPC UA. Jika Anda menggunakan fitur

stempel waktu dengan versi sebelumnya, pembaruan konfigurasi gagal. Untuk informasi selengkapnya, lihat [Perbarui versi AWS IoT SiteWise komponen](#).



Note

Pengaturan deadband berlaku ketika Anda telah memilih Berlangganan dalam pengaturan mode Pindai.

10. Pilih Simpan.

Konfigurasikan sumber OPC UA ()AWS CLI

Anda dapat menentukan sumber data OPC UA untuk gateway SiteWise Edge menggunakan file. AWS CLI Untuk melakukan ini, buat file JSON konfigurasi kemampuan OPC UA dan gunakan [update-gateway-capability-configuration](#) perintah untuk memperbarui konfigurasi gateway SiteWise Edge. Anda harus menentukan semua sumber OPC UA Anda dalam satu konfigurasi kemampuan.

MQTT-enabled, V3 gateway

Kemampuan ini memiliki namespace berikut.

- `iotsitewise:opcuacollector:3`

```
{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
        | "AES128_SHA256_RSA0AEP" | "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",
  "typeConversions": {
    "array": "JsonArray",
    "datetime": "ISO8601String"
  },
  "destination": {
    {
      "type": "MQTT"
    }
  },
  "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,
        "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
      }
    }
  ],

```

```

    "dataQuality": {
      "allowGoodQuality": true | false,
      "allowBadQuality": true | false,
      "allowUncertainQuality": true | false
    },
    "subscription": {
      "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
      "queueSize": integer,
      "publishingIntervalMilliseconds": integer,
      "snapshotFrequencyMilliseconds": integer
    }
  }
]
}
]
}

```

Classic streams, V2 gateway

Kemampuan ini memiliki namespace berikut.

- `iotsitewise:opcuacollector:2`

Sintaksis Permintaan

```

{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
| "AES128_SHA256_RSA0AEP" | "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",
    "typeConversions": {
      "array": "JsonArray",
      "datetime": "ISO8601String"
    },
    "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,
          "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
        },
        "dataQuality": {
          "allowGoodQuality": true | false,

```

```

        "allowBadQuality": true | false,
        "allowUncertainQuality": true | false
    },
    "subscription": {
        "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
        "queueSize": integer,
        "publishingIntervalMilliseconds": integer,
        "snapshotFrequencyMilliseconds": integer
    }
}
]
}
]
}

```

Isi permintaan

sources

Daftar struktur definisi sumber OPC UA yang masing-masing berisi informasi berikut:

name

Nama yang unik dan ramah untuk sumbernya.

endpoint

Struktur endpoint yang berisi informasi berikut:

certificateTrust

Struktur kebijakan kepercayaan sertifikat yang berisi informasi berikut:

type

Mode kepercayaan sertifikat untuk sumbernya. Pilih salah satu cara berikut:

- **TrustAny**— Gateway SiteWise Edge mempercayai sertifikat apa pun saat terhubung ke sumber OPC UA.
- **X509**— Gateway SiteWise Edge mempercayai sertifikat X.509 saat terhubung ke sumber OPC UA. Jika Anda memilih opsi ini, Anda harus menentukan `certificateBody` `certificateTrust`. Anda juga dapat menentukan `certificateChain` dalam `certificateTrust`.

certificateBody

(Opsional) Badan sertifikat X.509.

Bidang ini diperlukan jika Anda memilih X509 untuk type masukcertificateTrust.

certificateChain

(Opsional) Rantai kepercayaan untuk sertifikat X.509.

Bidang ini hanya digunakan jika Anda memilih X509 untuk type masukcertificateTrust.

endpointUri

Titik akhir lokal dari sumber OPC UA. Misalnya, titik akhir lokal Anda mungkin terlihat seperti `opc.tcp://203.0.113.0:49320`.

securityPolicy

Kebijakan keamanan yang digunakan sehingga Anda dapat mengamankan pesan yang dibaca dari sumber OPC UA. Pilih salah satu cara berikut:

- **NONE**— Gateway SiteWise Edge tidak mengamankan pesan dari sumber OPC UA. Kami menyarankan Anda memilih kebijakan keamanan yang berbeda. Jika Anda memilih opsi ini, Anda juga harus memilih **NONE** untuk `messageSecurityMode`.
- **BASIC256_SHA256**- Kebijakan `Basic256Sha256` keamanan.
- **AES128_SHA256_RSA0AEP**- Kebijakan `Aes128_Sha256_Rsa0aep` keamanan.
- **AES256_SHA256_RSAPSS**- Kebijakan `Aes256_Sha256_RsaPss` keamanan.
- **BASIC128_RSA15**— (Usang) Kebijakan `Basic128Rsa15` keamanan tidak digunakan lagi dalam spesifikasi OPC UA karena tidak lagi dianggap aman. Kami menyarankan Anda memilih kebijakan keamanan yang berbeda. Untuk informasi lebih lanjut, lihat [Profil SecurityPolicy — Basic128Rsa15](#).
- **BASIC256**— (Usang) Kebijakan `Basic256` keamanan tidak digunakan lagi dalam spesifikasi OPC UA karena tidak lagi dianggap aman. Kami menyarankan Anda memilih kebijakan keamanan yang berbeda. Untuk informasi lebih lanjut, lihat [SecurityPolicy — Basic256](#).

Important

Jika Anda memilih kebijakan keamanan selain **NONE**, Anda harus memilih **SIGN** atau **SIGN_AND_ENCRYPT** untuk `messageSecurityMode`. Anda juga harus

mengonfigurasi server sumber Anda untuk mempercayai gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Siapkan server OPC UA untuk mempercayai gateway AWS IoT SiteWise Edge](#).

messageSecurityMode

Mode keamanan pesan yang digunakan untuk mengamankan koneksi ke sumber OPC UA. Pilih salah satu cara berikut:

- **NONE**— Gateway SiteWise Edge tidak mengamankan koneksi ke sumber OPC UA. Kami menyarankan Anda memilih mode keamanan pesan yang berbeda. Jika Anda memilih opsi ini, Anda juga harus memilih **NONE** untuk **securityPolicy**.
- **SIGN**— Data dalam perjalanan antara gateway SiteWise Edge dan sumber OPC UA ditandatangani tetapi tidak dienkripsi.
- **SIGN_AND_ENCRYPT**— Data dalam perjalanan antara gateway dan sumber OPC UA ditandatangani dan dienkripsi.

Important

Jika Anda memilih mode keamanan pesan selain **NONE**, Anda harus memilih **securityPolicy** selain **NONE**. Anda juga harus mengonfigurasi server sumber Anda untuk mempercayai gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Siapkan server OPC UA untuk mempercayai gateway AWS IoT SiteWise Edge](#).

identityProvider

Struktur penyedia identitas yang berisi informasi berikut:

type

Jenis kredensial otentikasi yang diperlukan oleh sumber. Pilih salah satu cara berikut:

- **Anonymous**— Sumber tidak memerlukan otentikasi untuk terhubung.
- **Username**— Sumber membutuhkan nama pengguna dan kata sandi untuk terhubung. Jika Anda memilih opsi ini, Anda harus menentukan **usernameSecretArn** untuk **identityProvider**.

usernameSecretArn

(Optional) ARN rahasia. AWS Secrets Manager Gateway SiteWise Edge menggunakan kredensial otentikasi dalam rahasia ini ketika terhubung ke sumber ini. Anda harus melampirkan rahasia ke SiteWise konektor IoT gateway SiteWise Edge Anda untuk menggunakannya untuk otentikasi sumber. Untuk informasi selengkapnya, lihat [Konfigurasi otentikasi sumber data untuk Edge SiteWise](#).

Bidang ini diperlukan jika Anda memilih Username untuk type `masukidentityProvider`.

nodeFilterRules

Daftar struktur aturan filter node yang menentukan jalur aliran data OPC UA untuk dikirim ke AWS Cloud. Anda dapat menggunakan filter node untuk mengurangi waktu startup dan penggunaan CPU gateway SiteWise Edge Anda dengan hanya menyertakan jalur ke data yang Anda modelkan AWS IoT SiteWise. Secara default, gateway SiteWise Edge mengunggah semua jalur OPC UA kecuali yang dimulai dengan `/Server/`. Untuk menentukan filter node OPC UA, Anda dapat menggunakan jalur node `*` dan `**` karakter wildcard. Untuk informasi selengkapnya, lihat [Gunakan filter node OPC UA di SiteWise Edge](#).

Setiap struktur dalam daftar harus berisi informasi berikut:

action

Tindakan untuk aturan filter node ini. Anda dapat memilih opsi berikut:

- **INCLUDE**— Gateway SiteWise Edge hanya mencakup aliran data yang cocok dengan aturan ini.

definition

Struktur aturan filter node yang berisi informasi berikut:

type

Jenis jalur filter node untuk aturan ini. Anda dapat memilih opsi berikut:

- **OpcUaRootPath**— Gateway SiteWise Edge mengevaluasi jalur filter node ini terhadap root hierarki jalur OPC UA.

rootPath

Jalur filter node untuk mengevaluasi terhadap akar hierarki jalur OPC UA. Jalur ini harus dimulai dengan `/`.

measurementDataStreamPrefix

Sebuah string untuk ditambahkan ke semua aliran data dari sumber. Gateway SiteWise Edge menambahkan awalan ini ke semua aliran data dari sumber ini. Gunakan awalan aliran data untuk membedakan antara aliran data yang memiliki nama yang sama dari sumber yang berbeda. Setiap aliran data harus memiliki nama unik dalam akun Anda.

typeConversions

Jenis konversi yang tersedia untuk tipe data OPC UA yang tidak didukung. Setiap tipe data dikonversi menjadi string. Untuk informasi selengkapnya, lihat [Mengonversi tipe data yang tidak didukung](#).

array

Tipe data array sederhana yang dikonversi ke string. Anda dapat memilih opsi berikut:

- **JsonArray**— Menunjukkan bahwa Anda memilih untuk mengonversi tipe data array sederhana Anda menjadi string.

datetime

Tipe DateTime data yang dikonversi ke string. Anda dapat memilih opsi berikut:

- **ISO8601String**— Menunjukkan bahwa Anda memilih untuk mengonversi tipe data ISO 8601 menjadi string.

destination

Konfigurasi untuk tujuan tag OPC UA. Aliran klasik, v2 dan berkemampuan MQTT, gateway V3 memiliki konfigurasi yang berbeda untuk tujuan.

type

Jenis tujuan.

streamName— hanya untuk aliran Klasik, gateway V2

Nama sungai. Nama stream harus unik.

streamBufferSize— hanya untuk aliran Klasik, gateway V2

Ukuran buffer aliran. Ini penting untuk mengelola aliran data dari sumber OPC UA.

propertyGroups

(Opsional) Daftar grup properti yang menentukan deadband dan scanMode diminta oleh protokol.

name

Nama grup properti. Ini harus menjadi pengidentifikasi unik.

deadband

deadbandNilai mendefinisikan perubahan minimum dalam nilai titik data yang harus terjadi sebelum data dikirim ke cloud. Itu berisi informasi berikut:

type

Jenis deadband yang didukung. Anda dapat memilih opsi berikut:

- **ABSOLUTE**— Nilai tetap yang menentukan perubahan absolut minimum yang diperlukan untuk mempertimbangkan titik data yang cukup signifikan untuk dikirim ke cloud.
- **PERCENT**— Nilai dinamis yang menentukan perubahan minimum yang diperlukan sebagai persentase dari nilai titik data terakhir yang dikirim. Jenis deadband ini berguna ketika nilai data bervariasi secara signifikan dari waktu ke waktu.

value

Nilai deadband. typeKapanABSOLUTE, nilai ini adalah ganda tanpa unit.

typeKapanPERCENT, nilai ini adalah ganda antara 1 dan100.

eguMin

(Opsional) Unit teknik minimum saat menggunakan PERCENT deadband. Anda mengatur ini jika server OPC UA tidak memiliki unit teknik yang dikonfigurasi.

eguMax

(Opsional) Unit teknik maksimum saat menggunakan PERCENT deadband. Anda mengatur ini jika server OPC UA tidak memiliki unit teknik yang dikonfigurasi.

timeoutMilliseconds

Durasi dalam milidetik sebelum batas waktu. Minimal adalah100.

scanMode

scanModeStruktur yang berisi informasi berikut:

type

Jenis yang didukung dariscanMode. Nilai yang diterima adalah POLL danEXCEPTION.

rate

Interval pengambilan sampel untuk mode pemindaian.

timestampToReturn

Sumber stempel waktu. Anda dapat memilih opsi berikut:

- `SOURCE_TIME`— Menggunakan stempel waktu dari perangkat Anda.
- `SERVER_TIME`— Menggunakan stempel waktu dari server Anda.

Note

Gunakan `TimestampToReturn` dengan versi 2.5.0 atau yang lebih baru dari komponen kolektor SiteWise IoT OPC UA. Jika Anda menggunakan fitur ini dengan versi sebelumnya, pembaruan konfigurasi gagal. Untuk informasi selengkapnya, lihat [Perbarui versi AWS IoT SiteWise komponen](#).

nodeFilterRuleDefinitions

(Opsional) Daftar jalur simpul untuk disertakan dalam grup properti. Grup properti tidak dapat tumpang tindih. Jika Anda tidak menentukan nilai untuk bidang ini, grup berisi semua jalur di bawah root, dan Anda tidak dapat membuat grup properti tambahan.

`nodeFilterRuleDefinitions`Struktur berisi informasi berikut:

type

`OpcUaRootPath`adalah satu-satunya tipe yang didukung. Ini menentukan bahwa nilai `rootPath` adalah jalur relatif terhadap akar ruang penjelajahan OPC UA.

rootPath

Daftar yang dibatasi koma yang menentukan jalur (relatif terhadap root) untuk disertakan dalam grup properti.

Contoh konfigurasi kemampuan tambahan untuk aliran Klasik, gateway V2 ()AWS CLI

Contoh berikut mendefinisikan konfigurasi kemampuan gateway OPC UA SiteWise Edge dari payload yang disimpan dalam file JSON.

```
aws iotsitewise update-gateway-capability-configuration \  
--capability-namespace "iotsitewise:opcuacollector:2" \  

```

```
--capability-configuration file://opc-ua-configuration.json
```

Example : Konfigurasi sumber OPC UA

`opc-ua-configuration.json`File berikut mendefinisikan konfigurasi sumber OPC UA dasar dan tidak aman.

```
{
  "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 : Konfigurasi sumber OPC UA dengan grup properti yang ditentukan

`opc-ua-configuration.json`File berikut mendefinisikan konfigurasi sumber OPC UA dasar dan tidak aman dengan grup properti yang ditentukan.

```
{
  "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 : Konfigurasi sumber OPC UA dengan properti

Contoh JSON berikut untuk `opc-ua-configuration.json` mendefinisikan konfigurasi sumber OPC UA dengan properti berikut:

- Mempercayai sertifikat apa pun.
- Menggunakan kebijakan BASIC256 keamanan untuk mengamankan pesan.
- Menggunakan SIGN_AND_ENCRYPT mode untuk mengamankan koneksi.
- Menggunakan kredensi otentikasi yang disimpan dalam rahasia Secrets Manager.
- Memfilter aliran data kecuali yang jalurnya dimulai dengan `/WindFarm/2/WindTurbine/`.
- Menambahkan `/Washington` ke awal setiap jalur aliran data untuk membedakan antara "Wind Farm #2" dan "Wind Farm #2" di area lain.

```

{
  "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 : Konfigurasi sumber OPC UA dengan kepercayaan sertifikat

Contoh JSON berikut untuk `opc-ua-configuration.json` mendefinisikan konfigurasi sumber OPC UA dengan properti berikut:

- Mempercayai sertifikat X.509 yang diberikan.
- Menggunakan kebijakan BASIC256 keamanan untuk mengamankan pesan.
- Menggunakan SIGN_AND_ENCRYPT mode untuk mengamankan koneksi.

```

{
  "sources": [
    {
      "name": "Wind Farm #3",

```



```

    "endpoint": {
      "certificateTrust": {
        "type": "X509",
        "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBIDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC0lBTsBDb25zb2x1MRIw
EAYDVQQDEw1UZXRhbnQ2L2sYWMxH2AdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAwTC0lBTsBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ2L2sYWMx
H2AdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQEE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLygVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBZsWY6786m86gpEibb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
        "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBIDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC0lBTsBDb25zb2x1MRIw
EAYDVQQDEw1UZXRhbnQ2L2sYWMxH2AdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAwTC0lBTsBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ2L2sYWMx
H2AdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQEE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLygVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBZsWY6786m86gpEibb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
      },
      "endpointUri": "opc.tcp://203.0.113.2:49320",
      "securityPolicy": "BASIC256",
      "messageSecurityMode": "SIGN_AND_ENCRYPT",
      "identityProvider": {
        "type": "Anonymous"
      }
    },

```

```
        "nodeFilterRules": []
      },
      "measurementDataStreamPrefix": ""
    }
  ]
}
```

Siapkan server OPC UA untuk mempercayai gateway AWS IoT SiteWise Edge

Jika Anda memilih `messageSecurityMode` selain `None` saat mengonfigurasi sumber OPC UA Anda, Anda harus mengaktifkan server sumber Anda untuk mempercayai gateway AWS IoT SiteWise Edge. Gateway SiteWise Edge menghasilkan sertifikat yang mungkin diperlukan oleh server sumber Anda. Prosesnya bervariasi tergantung pada server sumber Anda. Untuk informasi selengkapnya, lihat dokumentasi untuk server Anda.

Prosedur berikut menguraikan langkah-langkah dasar.

Untuk mengaktifkan server OPC UA untuk mempercayai gateway SiteWise Edge

1. Buka antarmuka untuk mengonfigurasi server OPC UA Anda.
2. Masukkan nama pengguna dan kata sandi untuk administrator server OPC UA.
3. Temukan Klien Tepercaya di antarmuka, lalu pilih AWS IoT SiteWise Gateway Client.
4. Pilih Kepercayaan.

Mengekspor sertifikat klien OPC UA

Beberapa server OPC UA memerlukan akses ke file sertifikat klien OPC UA untuk mempercayai gateway SiteWise Edge. Jika ini berlaku untuk server OPC UA Anda, Anda dapat menggunakan prosedur berikut untuk mengekspor sertifikat klien OPC UA dari gateway SiteWise Edge. Kemudian, Anda dapat mengimpor sertifikat di server OPC UA Anda.

Untuk mengekspor file sertifikat klien OPC UA untuk sumber

1. Jalankan perintah berikut untuk mengubah ke direktori yang berisi file sertifikat. Ganti `sitewise-work` dengan jalur penyimpanan lokal untuk folder `aws.iot.SiteWiseEdgeCollectorOpcua` kerja Greengrass dan `source-name` ganti dengan nama sumber data.

Secara default, folder `/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua` kerja Greengrass ada di Linux dan seterusnya. `C:/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua` Microsoft Windows

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

2. Sertifikat klien OPC UA gateway SiteWise Edge untuk sumber ini ada di `aws-iot-opcua-client.pfx` file.

Jalankan perintah berikut untuk mengekspor sertifikat ke `.pem` file bernama `aws-iot-opcua-client-certificate.pem`.

```
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. Transfer file sertifikat, `aws-iot-opcua-client-certificate.pem`, dari gateway SiteWise Edge ke server OPC UA.

Untuk melakukannya, Anda dapat menggunakan perangkat lunak umum seperti `scp` program untuk mentransfer file menggunakan protokol SSH. Untuk informasi selengkapnya, lihat [Salinan aman](#) di Wikipedia.

Note

Jika gateway SiteWise Edge Anda berjalan di Amazon Elastic Compute Cloud (Amazon EC2) dan Anda menghubungkannya untuk pertama kalinya, Anda harus mengonfigurasi prasyarat untuk terhubung. Untuk informasi selengkapnya, lihat [Connect ke instans Linux menggunakan SSH](#) di Panduan EC2 Pengguna Amazon.

4. Impor file sertifikat, `aws-iot-opcua-client-certificate.pem`, di server OPC UA untuk mempercayai gateway SiteWise Edge. Langkah-langkah dapat bervariasi tergantung pada server sumber yang Anda gunakan. Konsultasikan dokumentasi untuk server.

Filter rentang konsumsi data dengan OPC UA

Anda dapat mengontrol cara Anda menelan data dengan sumber OPC UA dengan menggunakan mode pemindaian dan rentang deadband. Fitur-fitur ini memungkinkan Anda mengontrol jenis data

apa yang akan dikonsumsi, dan bagaimana serta kapan server dan gateway SiteWise Edge Anda bertukar informasi ini.

Mengumpulkan atau menyaring data berdasarkan kualitas

Anda dapat mengonfigurasi pengaturan kualitas data Anda untuk mengontrol data apa yang dikumpulkan dari sumber OPC UA. Sumber data mencakup peringkat kualitas sebagai metadata saat mengirimkannya. Anda dapat memilih salah satu atau semua opsi berikut:

- Good
- Bad
- Uncertain

Menangani NaN atau nilai null

SiteWise Edge mendukung pengumpulan dan penanganan NaN dan nilai null.

- NaN (Bukan Angka): Merupakan hasil numerik yang tidak terdefinisi atau tidak dapat direpresentasikan.
- Null: Menunjukkan data yang hilang.

Kolektor IoT SiteWise OPC UA menangkap nilai NaN dan Null dengan kualitas BURUK atau TIDAK PASTI. Nilai-nilai khusus ini ditulis ke aliran lokal, memungkinkan pengumpulan data yang lebih komprehensif.

Kontrol frekuensi pengumpulan data dengan mode Pindai

Anda dapat mengonfigurasi mode pemindaian OPC UA Anda untuk mengontrol cara Anda mengumpulkan data dari sumber OPC UA Anda. Anda dapat memilih mode berlangganan atau polling.

- Mode berlangganan — Sumber OPC UA mengumpulkan data untuk dikirim ke gateway SiteWise Edge Anda pada frekuensi yang ditentukan oleh tingkat pemindaian Anda. Server hanya mengirim data ketika nilai telah berubah, jadi ini adalah frekuensi maksimum gateway SiteWise Edge Anda menerima data.
- Mode polling - Gateway SiteWise Edge Anda menjajak pendapat sumber OPC UA pada frekuensi yang ditetapkan yang ditentukan oleh tingkat pemindaian Anda. Server mengirimkan data terlepas

dari apakah nilainya telah berubah, sehingga gateway SiteWise Edge Anda selalu menerima data pada interval ini.

 Note

Opsi mode polling mengesampingkan pengaturan deadband Anda untuk sumber ini.

Filter konsumsi data OPC UA dengan rentang deadband

Anda dapat menerapkan deadband ke grup properti sumber OPC UA Anda untuk memfilter dan membuang data tertentu alih-alih mengirimkannya ke Cloud. AWS Deadband menentukan jendela fluktuasi yang diharapkan dalam nilai data yang masuk dari sumber OPC UA Anda. Jika nilai termasuk dalam jendela ini, server OPC UA Anda tidak akan mengirimkannya ke AWS Cloud. Anda dapat menggunakan pemfilteran deadband untuk mengurangi jumlah data yang Anda proses dan kirim ke Cloud. AWS Untuk mempelajari cara menyiapkan sumber OPC UA untuk gateway SiteWise Edge Anda, lihat [Sumber data OPC UA untuk gateway AWS IoT SiteWise Edge](#).

 Note

Server Anda menghapus semua data yang ada di dalam jendela yang ditentukan oleh deadband Anda. Anda tidak dapat memulihkan data yang dibuang ini.

Jenis deadband

Anda dapat menentukan dua jenis deadband untuk grup properti server OPC UA Anda. Ini memungkinkan Anda memilih berapa banyak data yang dikirim ke AWS Cloud, dan berapa banyak yang dibuang.

- **Persentase** - Anda menentukan jendela menggunakan persentase fluktuasi yang diharapkan dalam nilai pengukuran. Server menghitung jendela yang tepat dari persentase ini, dan mengirimkan data ke AWS Cloud yang melebihi jatuh di luar jendela. Misalnya, menentukan nilai deadband 2% pada sensor dengan kisaran dari -100 derajat Fahrenheit hingga +100 derajat Fahrenheit memberi tahu server untuk mengirim data ke Cloud ketika nilainya berubah 4 derajat Fahrenheit atau lebih. AWS

 Note

Anda dapat secara opsional menentukan nilai deadband minimum dan maksimum untuk jendela ini jika server sumber Anda tidak menentukan unit teknik. Jika rentang unit teknik tidak disediakan, server OPC UA default ke rentang penuh tipe data pengukuran.

- **Absolute** — Anda menentukan jendela menggunakan unit yang tepat. Misalnya, menentukan nilai deadband 2 pada sensor memberitahu server untuk mengirim data ke AWS Cloud ketika nilainya berubah setidaknya 2 unit. Anda dapat menggunakan deadbanding absolut untuk lingkungan dinamis di mana fluktuasi diharapkan secara teratur selama operasi normal.

Batas waktu deadband

Anda dapat secara opsional mengonfigurasi pengaturan batas waktu deadband. Setelah batas waktu ini, server OPC UA mengirimkan nilai pengukuran saat ini meskipun berada dalam fluktuasi deadband yang diharapkan. Anda dapat menggunakan pengaturan batas waktu untuk memastikan bahwa AWS IoT SiteWise menelan aliran data yang stabil setiap saat, bahkan ketika nilai tidak melebihi jendela deadband yang ditentukan.

Gunakan filter node OPC UA di SiteWise Edge

Saat Anda menentukan sumber data OPC UA untuk gateway SiteWise Edge, Anda dapat menentukan filter node. Filter node memungkinkan Anda membatasi jalur aliran data mana yang dikirim gateway SiteWise Edge ke cloud. Anda dapat menggunakan filter node untuk mengurangi waktu startup dan penggunaan CPU gateway SiteWise Edge Anda dengan hanya menyertakan jalur ke data yang Anda modelkan AWS IoT SiteWise. Secara default, gateway SiteWise Edge mengunggah semua jalur OPC UA kecuali yang dimulai dengan `/Server/`. Anda dapat menggunakan karakter `*` dan `**` wildcard di filter node Anda untuk menyertakan beberapa jalur aliran data dengan satu filter. Untuk mempelajari cara menyiapkan sumber OPC UA untuk gateway SiteWise Edge Anda, lihat [Sumber data OPC UA untuk gateway AWS IoT SiteWise Edge](#).

 Note

AWS IoT SiteWise restart gateway SiteWise Edge Anda setiap kali Anda menambahkan atau mengedit sumber. Gateway SiteWise Edge Anda tidak akan menelan data saat memperbarui konfigurasi sumber. Waktu untuk memulai ulang gateway SiteWise Edge Anda tergantung pada jumlah tag pada sumber gateway SiteWise Edge Anda. Waktu restart dapat berkisar

dari beberapa detik (untuk gateway SiteWise Edge dengan beberapa tag) hingga beberapa menit (untuk gateway SiteWise Edge dengan banyak tag).

Tabel berikut mencantumkan wildcard yang dapat Anda gunakan untuk memfilter sumber data OPC UA.

OPC UA node filter wildcard

Wildcard	Deskripsi
*	Cocokkan satu level dalam jalur aliran data.
**	Cocokkan beberapa level di jalur aliran data.

Note

Jika Anda mengonfigurasi sumber dengan filter luas dan kemudian mengubah sumber untuk menggunakan filter yang lebih ketat, AWS IoT SiteWise berhenti menyimpan data yang tidak cocok dengan filter baru.

Example : Skenario menggunakan filter node

Pertimbangkan aliran data hipotetis berikut:

- /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

Menggunakan aliran data sebelumnya, Anda dapat menentukan filter node untuk membatasi data apa yang akan disertakan dari sumber OPC UA Anda.

- Untuk memilih semua node dalam contoh ini, gunakan `/` atau `**/`. Anda dapat menyertakan beberapa direktori atau folder dengan `**` karakter wildcard.
- Untuk memilih semua aliran PLC data, gunakan `/**/*.PLC*` atau `**/PLC*`.
- Untuk memilih semua penghitung dalam contoh ini, gunakan `/**/*.Counter*` atau `/**/*.Counter*`.
- Untuk memilih semua penghitung dari Line 2, gunakan `**/Line 2/Counter*`.

Mengonversi tipe data yang tidak didukung

Secara opsional aktifkan konversi tipe data AWS IoT SiteWise untuk array sederhana dan tipe DateTime data. AWS IoT SiteWise tidak mendukung semua tipe data OPC UA. Ketika Anda mengirim data yang tidak didukung ke aliran AWS IoT Greengrass data Anda, data tersebut hilang. Namun, dengan mengonversi tipe data asli yang tidak didukung menjadi string, Anda dapat menelan data AWS IoT SiteWise daripada membuangnya. AWS IoT SiteWise serialisasi data Anda yang dikonversi sehingga nantinya Anda dapat menggunakan fungsi Anda sendiri untuk mengonversi string kembali ke tipe data aslinya di hilir, jika diperlukan.

Anda dapat memperbarui pengaturan konversi tipe data untuk sumber data kapan saja dan setiap sumber data dapat memiliki pengaturannya sendiri.

Saat Anda menambahkan sumber data di Konsol AWS IoT SiteWise, ada dua kotak centang di bawah Konversi tipe data di Konfigurasi Lanjutan. Anda dapat menunjukkan tipe data mana yang akan dikonversi ke string.

Selain itu, kolektor IoT SiteWise OPC UA dapat menerima NaN atau nilai nol di tepi.

- Mengkonversi nilai array dengan tipe data sederhana ke string JSON
- Konversi DateTime nilai ke string ISO 8601

Prasyarat

- Gunakan kolektor [SiteWise IoT OPC](#) UA versi 2.5.0 atau yang lebih baru.

Batasan

Ini adalah batasan untuk konversi tipe data OPC UA ke string di AWS IoT SiteWise

- Konversi tipe data yang kompleks tidak didukung.

- Batas string setelah konversi adalah 1024 byte. Jika string lebih panjang dari 1024 byte, string ditolak oleh. AWS IoT SiteWise

Konfigurasi otentikasi sumber data untuk Edge SiteWise

Jika server OPC UA Anda memerlukan kredensial otentikasi untuk terhubung, Anda dapat menggunakannya AWS Secrets Manager untuk membuat dan menyebarkan rahasia ke gateway Edge Anda. SiteWise AWS Secrets Manager mengenkripsi rahasia pada perangkat untuk menjaga nama pengguna dan kata sandi Anda tetap aman sampai Anda perlu menggunakannya. Untuk informasi selengkapnya tentang komponen manajer AWS IoT Greengrass rahasia, lihat [Manajer rahasia](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Untuk informasi tentang mengelola akses ke rahasia Secrets Manager, lihat:

- [Siapa yang memiliki izin untuk AWS Secrets Manager rahasia Anda.](#)
- [Menentukan apakah permintaan diizinkan atau ditolak dalam akun.](#)

Langkah 1: Buat rahasia otentikasi sumber

Anda dapat menggunakan AWS Secrets Manager untuk membuat rahasia otentikasi untuk sumber data Anda. Secara rahasia, tentukan **username** dan pasangan **password** nilai kunci yang berisi detail otentikasi untuk sumber data Anda.

Untuk membuat rahasia (konsol)

1. Navigasikan ke [konsol AWS Secrets Manager](#) tersebut.
2. Pilih Simpan rahasia baru.
3. Di bawah Tipe rahasia, pilih Jenis rahasia lainnya.
4. Di bawah pasangan kunci/nilai, lakukan hal berikut:
 1. Di kotak input pertama, masukkan **username** dan di kotak input kedua masukkan nama pengguna.
 2. Pilih Tambahkan baris.
 3. Di kotak input pertama, masukkan **password** dan di kotak input kedua masukkan kata sandi.
5. Untuk kunci Enkripsi, pilih aws/secretsmanager, lalu pilih Berikutnya.
6. Di Store halaman rahasia baru, masukkan nama Rahasia.

7. (Opsional) Masukkan Deskripsi yang membantu Anda mengidentifikasi rahasia ini, lalu pilih Berikutnya.
8. (Opsional) Di Menyimpan halaman rahasia baru, nyalakan Rotasi otomatis. Untuk informasi selengkapnya, lihat [Memutar rahasia](#) di Panduan AWS Secrets Manager Pengguna.
9. Tentukan jadwal rotasi.
10. Pilih fungsi Lambda yang dapat memutar rahasia ini, lalu pilih Berikutnya.
11. Tinjau konfigurasi rahasia Anda, lalu pilih Store.

Untuk mengotorisasi gateway SiteWise Edge Anda untuk berinteraksi AWS Secrets Manager, peran IAM untuk gateway SiteWise Edge Anda harus mengizinkan tindakan tersebut `secretsmanager:GetSecretValue`. Anda dapat menggunakan perangkat inti Greengrass untuk mencari kebijakan IAM. Untuk informasi selengkapnya tentang memperbarui kebijakan IAM, lihat [Mengedit kebijakan IAM](#) di AWS Identity and Access Management Panduan Pengguna.

Example kebijakan

Ganti `secret-arn` dengan Nama Sumber Daya Amazon (ARN) dari rahasia yang Anda buat di langkah sebelumnya. Untuk informasi selengkapnya tentang cara mendapatkan ARN rahasia, lihat [Temukan rahasia AWS Secrets Manager di AWS Secrets Manager](#) Panduan Pengguna.

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

Langkah 2: Terapkan rahasia ke perangkat gateway SiteWise Edge Anda

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menyebarkan rahasia ke gateway SiteWise Edge Anda.

Untuk menyebarkan rahasia (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Gateway.
3. Dari daftar Gateways, pilih gateway SiteWise Edge target.
4. Di bagian konfigurasi Gateway, pilih tautan perangkat inti Greengrass untuk membuka inti yang terkait dengan gateway AWS IoT Greengrass Edge. SiteWise
5. Di panel navigasi, pilih Deployment.
6. Pilih penyebaran target, lalu pilih Revisi.
7. Pada halaman Tentukan target, pilih Selanjutnya.
8. Pada halaman Pilih komponen, di bagian Komponen publik, matikan Tampilkan hanya komponen yang dipilih.
9. Cari dan pilih `aws.greengrass.SecretManager` komponen, dan kemudian pilih Berikutnya.
10. Dari daftar Komponen yang dipilih, pilih `aws.greengrass.SecretManager` komponen, dan kemudian pilih Configure component.
11. Di bidang Configuration to merge, tambahkan objek JSON berikut.

Note

Ganti `secret-arn` dengan ARN rahasia yang Anda buat di langkah sebelumnya. Untuk informasi selengkapnya tentang cara mendapatkan ARN rahasia, lihat [Temukan rahasia AWS Secrets Manager di AWS Secrets Manager](#) Panduan Pengguna.

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

12. Pilih Konfirmasi.
13. Pilih Berikutnya.
14. Pada halaman Konfigurasi pengaturan lanjutan, pilih Berikutnya.
15. Tinjau konfigurasi penerapan Anda, lalu pilih Deploy.

Langkah 3: Tambahkan konfigurasi otentikasi

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menambahkan konfigurasi otentikasi ke gateway SiteWise Edge Anda.

Untuk menambahkan konfigurasi otentikasi (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Dari daftar Gateways, pilih gateway SiteWise Edge target.
3. Dari daftar Sumber data, pilih sumber data target, lalu pilih Edit.
4. Pada halaman Tambahkan sumber data, pilih Konfigurasi lanjutan.
5. Untuk konfigurasi Otentikasi, pilih rahasia yang Anda gunakan di langkah sebelumnya.
6. Pilih Simpan.

Sumber data mitra di gateway SiteWise Edge

Saat menggunakan gateway AWS IoT SiteWise Edge, Anda dapat menghubungkan sumber data mitra ke gateway SiteWise Edge Anda dan menerima data dari mitra di gateway SiteWise Edge dan AWS cloud Anda. Sumber data mitra ini adalah AWS IoT Greengrass komponen yang dikembangkan dalam kemitraan antara AWS dan mitra. Saat Anda menambahkan sumber data mitra, AWS IoT SiteWise akan membuat komponen ini dan menerapkannya di gateway SiteWise Edge Anda.

Note

Anda dapat menambahkan satu sumber data untuk setiap mitra di setiap gateway.

Untuk menambahkan sumber data mitra, lakukan hal berikut:

1. [Tambahkan sumber data mitra di SiteWise Edge](#)

2. Buka portal web mitra, jika berlaku, dan konfigurasi sumber data mitra sehingga terhubung ke gateway SiteWise Edge.

Topik

- [Keamanan](#)
- [Siapkan Docker di gateway SiteWise Edge Anda](#)
- [Tambahkan sumber data mitra di SiteWise Edge](#)
- [SiteWise Opsi sumber data mitra gateway tepi](#)

Keamanan

Sebagai bagian dari [Model Tanggung Jawab Bersama](#) antara AWS, pelanggan kami, dan mitra kami, berikut ini menjelaskan siapa yang bertanggung jawab atas berbagai aspek keamanan:

Tanggung jawab pelanggan

- Memeriksa pasangan.
- Mengkonfigurasi akses jaringan yang diberikan kepada mitra.
- Pemantauan untuk penggunaan yang wajar dari sumber daya mesin gateway SiteWise Edge (CPU, memori, dan sistem file).

AWS tanggung jawab

- Mengisolasi mitra dari sumber daya AWS cloud pelanggan kecuali yang dibutuhkan oleh mitra. Dalam hal ini, AWS IoT SiteWise konsumsi.
- Membatasi solusi mitra untuk penggunaan yang wajar dari sumber daya mesin gateway SiteWise Edge (CPU dan memori).

Tanggung jawab mitra

- Menggunakan default yang aman.
- Menjaga solusi tetap aman dari waktu ke waktu melalui tambalan dan pembaruan lain yang sesuai.
- Menjaga kerahasiaan data pelanggan.

Siapkan Docker di gateway SiteWise Edge Anda

AWS IoT SiteWise menyediakan Docker gambar yang memungkinkan Anda menjalankan aplikasi SiteWise Edge di berbagai platform dan lingkungan. DockerGambar ini merangkum semua

komponen dan dependensi yang diperlukan untuk mengumpulkan, memproses, dan mengirim data dari peralatan industri Anda ke Cloud. AWS Dengan menggunakan image Docker, Anda dapat menerapkan dan menjalankan aplikasi SiteWise Edge pada host yang kompatibel dengan Docker, seperti server, perangkat edge, atau layanan kontainer berbasis cloud.

Untuk menambahkan sumber data mitra, [Docker Engine](#) 1.9.1 atau yang lebih baru harus diinstal pada perangkat lokal Anda.

 Note

Versi 20.10 adalah versi terbaru yang diverifikasi untuk bekerja dengan perangkat lunak gateway SiteWise Edge.

Verifikasi Docker sudah diinstal

Untuk memverifikasi Docker diinstal, jalankan perintah berikut dari terminal yang terhubung ke gateway SiteWise Edge Anda:

```
docker info
```

Jika perintah mengembalikan `docker is not recognized` hasil, atau versi yang lebih lama diinstal, [Instal Docker Engine](#) sebelum melanjutkan. Docker

Siapkan Docker

Pengguna sistem yang menjalankan komponen Docker kontainer harus memiliki izin root atau administrator, atau Anda harus mengonfigurasi Docker untuk menjalankannya sebagai pengguna non-root atau non-administrator.

Pada perangkat Linux, Anda harus menambahkan `ggc_user` pengguna ke `docker` grup untuk memanggil Docker perintah tanpa `sudo`.

Untuk menambah `ggc_user`, atau pengguna non-root yang Anda gunakan untuk menjalankan komponen Docker kontainer, ke `docker` grup, jalankan perintah berikut:

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

Untuk informasi selengkapnya, lihat [langkah-langkah pasca-instalasi Linux untuk Docker Engine](#).

Tambahkan sumber data mitra di SiteWise Edge

Untuk menghubungkan sumber data mitra ke gateway SiteWise Edge Anda, tambahkan sebagai sumber data. Saat Anda menambahnya sebagai sumber data, AWS IoT SiteWise akan menyebarkan AWS IoT Greengrass komponen pribadi ke gateway SiteWise Edge Anda.

Prasyarat

Untuk menambahkan sumber data mitra, Anda harus melakukan hal berikut:

- Untuk EasyEdge dan CloudRail, buat akun dengan mitra, lalu ikat akun.
- [Siapkan Docker di gateway SiteWise Edge Anda](#)

Buat gateway SiteWise Edge dengan sumber data mitra

Jika Anda ingin membuat gateway SiteWise Edge baru, selesaikan langkah-langkahnya [Buat gateway SiteWise Edge yang dihosting sendiri](#). Setelah Anda membuat gateway SiteWise Edge, ikuti langkah-langkah [Tambahkan sumber data mitra ke gateway SiteWise Edge yang ada](#) untuk menambahkan sumber data mitra.

Tambahkan sumber data mitra ke gateway SiteWise Edge yang ada

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway SiteWise Edge yang ingin Anda sambungkan dengan sumber data mitra.
4. Di bawah Sumber data, pilih Tambahkan sumber data.
5. Pada layar Tambahkan sumber data, pilih tipe Sumber, untuk memilih mitra yang menghubungkan gateway SiteWise Edge Anda. Setiap sumber data memiliki opsi konfigurasi sendiri. Ada dua kategori sumber data: sumber dan AWS sumber Mitra.

Menggunakan sumber data mitra, Anda dapat memilih satu sumber per gateway. Untuk daftar opsi integrasi mitra sumber data, lihat [SiteWise Opsi sumber data mitra gateway tepi](#). Perhatikan bahwa Anda dapat menambahkan hingga 100 sumber data OPC UA (AWS sumber). Untuk memulai dengan sumber data OPC UA, lihat [Sumber data OPC UA untuk gateway AWS IoT SiteWise Edge](#).

6. Masukkan nama untuk sumbernya.
7. Pilih tab sumber data Anda di bawah ini dan ikuti prosedur konfigurasi.

CloudRail

Sebagian besar CloudRail konfigurasi dilakukan di CloudRail portal setelah menyimpan sumber data untuk gateway SiteWise Edge Anda. Namun, otorisasi koneksi diperlukan.

Note

CloudRailKoneksi hanya tersedia di Linux.

1. [Buat CloudRail akun](#) untuk memulai dengan menghubungkan ke AWS IoT SiteWise.
2. Pastikan Docker diinstal pada gateway Anda. Untuk informasi selengkapnya, lihat [Siapkan Docker di gateway SiteWise Edge Anda](#).
3. Baca Perjanjian Akses dan Penyebaran Otorisasi, lalu pilih Otorisasi. Memeriksa kotak memberi AWS mitra akses ke sumber data Anda dan memungkinkan AWS untuk menerapkan pada komponen mitra.

Note

Awalan Pengukuran — opsional diatur dalam CloudRail portal Anda.

Note

Perangkat lunak mitra dikembangkan, dipelihara, dan didukung oleh AWS mitra. AWS tidak bertanggung jawab atas antarmuka, konfigurasi, atau perangkat lunak.

Untuk informasi selengkapnya, lihat [CloudRail](#).

EasyEdge

Sebagian besar EasyEdge konfigurasi dilakukan di EasyEdge portal setelah menyimpan sumber data untuk gateway SiteWise Edge Anda. Namun, otorisasi koneksi diperlukan.

 Note

EasyEdgeKoneksi hanya tersedia di Linux.

1. [Buat EasyEdge akun](#) untuk memulai dengan menghubungkan ke AWS IoT SiteWise.
2. Pastikan Docker diinstal pada gateway Anda. Untuk informasi selengkapnya, lihat [Siapkan Docker di gateway SiteWise Edge Anda](#).
3. Baca Perjanjian Akses dan Penyebaran Otorisasi, lalu pilih Otorisasi. Memeriksa kotak memberi AWS mitra akses ke sumber data Anda dan memungkinkan AWS untuk menerapkan pada komponen mitra.

 Note

Awalan Pengukuran — opsional diatur dalam EasyEdge portal Anda.

 Note

Perangkat lunak mitra dikembangkan, dipelihara, dan didukung oleh AWS mitra. AWS tidak bertanggung jawab atas antarmuka, konfigurasi, atau perangkat lunak.

Untuk informasi selengkapnya, lihat [EasyEdge](#).

Tepi Lakmus

Anda dapat mengaktifkan Litmus konfigurasi dengan dua cara. Aktifkan Litmus Edge langsung melalui AWS IoT SiteWise penggunaan informasi dari Litmus Edge Manager portal. Atau, Anda dapat mengaktifkan secara manual Litmus Edge untuk AWS IoT SiteWise melalui Litmus Edge Manager.

 Note

Litmus EdgeKoneksi hanya tersedia di Linux.

Untuk mengaktifkan menggunakan kode Litmus Edge aktivasi pada AWS IoT SiteWise

Gunakan prosedur ini saat menambahkan sumber Litmus Edge data dengan kode Litmus Edge aktivasi pada file Konsol AWS IoT SiteWise.

1. Pilih Aktifkan sekarang menggunakan kode. Opsi konfigurasi tambahan muncul.
2. Masukkan tombol Litmus Edge Manager to connect Litmus Edge ke gateway SiteWise Edge Anda. Untuk informasi selengkapnya, lihat [Langkah 3a: Mengatur Titik Akhir Manajemen Data dan Perangkat](#) dalam dokumentasi. Litmus Edge Manager
3. Berikan kode Litmus Edge Manager aktivasi untuk Litmus Edge diaktifkan AWS IoT SiteWise
4. Secara opsional, AWS IoT SiteWise berikan sertifikat Litmus Edge Manager CA. Sertifikat Litmus Edge mencegah pengaktifan pada yang tidak sah Litmus Edge Manager.
5. Pastikan Docker diinstal pada gateway Anda. Untuk informasi selengkapnya, lihat [Siapkan Docker di gateway SiteWise Edge Anda](#).

 Note

AWS IoT SiteWise menyebarkan aplikasi mitra sebagai wadah Docker. Aplikasi ini digunakan dengan NET_ADMIN kemampuan sehingga wadah Litmus Edge Docker dapat dikelola melalui. Litmus Edge Manager Litmus Edgememerlukan akses istimewa ini untuk berjalan di perangkat Anda. Untuk informasi selengkapnya tentang persyaratan Litmus Edge Docker, lihat [Instalasi Docker](#) di QuickStart Panduan dalam dokumentasi Litmus Edge.

6. Baca Perjanjian Akses dan Penyebaran Otorisasi, lalu pilih Otorisasi. Memeriksa kotak memberi AWS mitra akses ke sumber data Anda dan memungkinkan AWS untuk menerapkan pada komponen mitra.

Untuk mengaktifkan secara manual melalui Litmus Edge

1. Pilih Aktifkan nanti Litmus Edge.

2. Pastikan Docker diinstal pada gateway Anda. Untuk informasi selengkapnya, lihat [Siapkan Docker di gateway SiteWise Edge Anda](#).

 Note

AWS IoT SiteWise menyebarkan aplikasi mitra sebagai wadah Docker. Aplikasi ini digunakan dengan NET_ADMIN kemampuan sehingga wadah Litmus Edge Docker dapat dikelola melalui Litmus Edge Manager. Litmus Edgememerlukan akses istimewa ini untuk berjalan di perangkat Anda. Untuk informasi selengkapnya tentang persyaratan Litmus Edge Docker, lihat [Instalasi Docker](#) di QuickStart Panduan dalam dokumentasi Litmus Edge.

3. Baca Perjanjian Akses dan Penyebaran Otorisasi, lalu pilih Otorisasi. Memeriksa kotak memberi AWS mitra akses ke sumber data Anda dan memungkinkan AWS untuk menerapkan pada komponen mitra.
4. Setelah penerapan selesai, ikuti petunjuk [Access the Litmus Edge Web UI](#) dalam dokumentasi. Litmus Edge QuickStart Guide

 Note

Perangkat lunak mitra dikembangkan, dipelihara, dan didukung oleh AWS mitra. AWS tidak bertanggung jawab atas antarmuka, konfigurasi, atau perangkat lunak.

Untuk informasi selengkapnya, lihat [Litmus Edge](#).

8. Pilih Simpan.

SiteWise Opsi sumber data mitra gateway tepi

AWS IoT SiteWise memungkinkan Anda untuk menghubungkan dan menelan data dari berbagai sumber data mitra, seperti peralatan industri, sensor, dan sistem pihak ketiga lainnya. Untuk menghubungkan sumber data mitra, Anda perlu mengikuti beberapa langkah, termasuk mengonfigurasi sumber data untuk mengirim data AWS IoT SiteWise, menyiapkan izin dan otentikasi yang diperlukan, dan memetakan data ke model aset Anda. Proses ini memastikan bahwa data mitra Anda terintegrasi dengan mulus ke AWS IoT SiteWise lingkungan Anda, memungkinkan Anda untuk memantau dan menganalisisnya bersama sumber data Anda yang lain.

Bagian ini mencantumkan mitra yang tersedia untuk integrasi sumber data pihak ketiga di gateway SiteWise Edge. Gunakan informasi di bawah ini untuk mengonfigurasi sumber data mitra.

 Note

Anda dapat menambahkan satu sumber data untuk setiap mitra di setiap gateway

CloudRail

Portal:

<https://devices.cloudrail.com/>

Persyaratan

Untuk informasi lebih lanjut tentang CloudRail persyaratan, lihat [FAQ](#) di situs web. CloudRail CloudRaildokumentasi:

[Komputasi Tepi: SiteWise Tepi](#)

EasyEdge

Portal:

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

Persyaratan

[EasyEdgepersyaratan](#) — Informasi tentang EasyEdge persyaratan, termasuk titik akhir dan port yang diperlukan untuk mengkonfigurasi firewall. Catatan: Anda memerlukan EasyEdge akun untuk mengakses dokumentasi ini.

EasyEdgedokumentasi:

[EasyEdgeuntuk AWS](#)

Litmus Edge

Akses keLitmus Edge Manager:

Untuk mengakses Litmus Edge, siapkan akun [Lakmus Edge Manager](#).

Persyaratan

[Litmus EdgePersyaratan](#) - Konfigurasi yang disarankan dan persyaratan sistem untuk diterapkanLitmus Edge.

Litmusdokumentasi:

- [Integrasi ke AWS IoT SiteWise](#)
- [Dokumentasi Litmus Edge](#)

AWS IoT Greengrass komponen untuk AWS IoT SiteWise Edge

SiteWise Edge menggunakan AWS IoT Greengrass komponen untuk mengumpulkan, memproses, dan mengirimkan data industri di tepi. Komponen-komponen ini bekerja sama untuk memungkinkan pemrosesan data lokal dan integrasi tanpa batas dengan layanan AWS IoT SiteWise cloud.

Penerbit IoT SiteWise

Komponen SiteWise penerbit IoT (`aws.iot.SiteWiseEdgePublisher`) bertanggung jawab untuk:

- Mentransmisikan data yang dikumpulkan dengan aman ke layanan cloud AWS IoT SiteWise
- Mengelola buffering data dan percobaan ulang selama masalah konektivitas

Untuk informasi selengkapnya tentang mengonfigurasi penerbit untuk SiteWise Edge, lihat [Konfigurasi AWS IoT SiteWise komponen penerbit](#). Dan, untuk informasi selengkapnya tentang komponen penerbit, lihat penerbit [SiteWise IoT](#) di AWS IoT Greengrass Version 2 Panduan Pengembang.

Prosesor IoT SiteWise

Komponen SiteWise prosesor IoT (`aws.iot.SiteWiseEdgeProcessor`) melakukan tugas-tugas berikut:

- Menjalankan transformasi data dan perhitungan di tepi
- Menerapkan definisi dan perhitungan properti aset secara lokal
- Mengurangi volume data dengan mengumpulkan atau memfilter data sebelum transmisi

Untuk informasi selengkapnya tentang komponen prosesor, lihat [SiteWise Prosesor IoT](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Kolektor IoT SiteWise OPC UA

Komponen IoT SiteWise OPC UA collector (`aws.iot.SiteWiseEdgeCollectorOpcua`) dirancang untuk:

- Connect ke server OPC UA di lingkungan industri
- Kumpulkan data dari sumber data OPC UA secara efisien
- Ubah data OPC UA menjadi format yang kompatibel AWS IoT SiteWise

Untuk informasi selengkapnya tentang komponen kolektor OPC UA, lihat kolektor [SiteWise IoT OPC UA](#) di AWS IoT Greengrass Version 2 Panduan Pengembang.

Simulator sumber data IoT SiteWise OPC UA

Komponen simulator sumber data IoT SiteWise OPC UA (`aws.iot.SiteWiseEdgeOpcuaDataSourceSimulator`) menyediakan fungsionalitas berikut:

- Memulai server OPC UA lokal yang menghasilkan data sampel
- Mensimulasikan sumber data yang dapat dibaca oleh komponen kolektor AWS IoT SiteWise OPC UA pada gateway AWS IoT SiteWise
- Memungkinkan eksplorasi AWS IoT SiteWise fitur menggunakan data sampel yang dihasilkan

Komponen ini sangat berguna untuk tujuan pengujian dan pengembangan, memungkinkan Anda untuk mensimulasikan sumber data industri tanpa memerlukan peralatan fisik.

Untuk informasi selengkapnya tentang komponen simulasi sumber data, lihat simulator [sumber data SiteWise IoT OPC UA](#) di AWS IoT Greengrass Version 2 Panduan Pengembang.

AWS IoT Greengrass Komponen-komponen ini berfungsi untuk mengaktifkan fungsionalitas SiteWise Edge. SiteWise Penerbit IoT memastikan data dikirim dengan andal ke cloud, SiteWise prosesor IoT menangani komputasi lokal dan pengoptimalan data, dan pengumpul IoT OPC UA memfasilitasi integrasi dengan protokol industri umum. SiteWise

Note

Untuk menggunakan komponen ini, Anda harus menginstal AWS IoT Greengrass V2 atau nanti di perangkat edge Anda. Konfigurasi yang tepat dari setiap komponen penting untuk kinerja SiteWise Edge yang optimal.

Filter aset di gateway SiteWise Edge

Anda dapat menggunakan pemfilteran tepi untuk mengelola aset Anda secara lebih efisien dengan mengirimkan hanya sebagian aset ke gateway SiteWise Edge tertentu untuk digunakan dalam pemrosesan data. Jika aset Anda diatur dalam struktur pohon, atau orang tua-anak, Anda dapat menyiapkan kebijakan IAM yang dilampirkan ke peran IAM gateway SiteWise Edge yang hanya mengizinkan akar pohon, atau induk, dan anak-anaknya dikirim ke gateway Edge tertentu. SiteWise

Note

Jika Anda mengatur aset yang ada ke dalam struktur pohon, setelah Anda membuat struktur, masuk ke setiap aset yang ada yang Anda tambahkan ke struktur dan pilih Edit lalu pilih Simpan untuk memastikan AWS IoT SiteWise mengenali struktur baru.

Siapkan penyaringan tepi

Siapkan pemfilteran tepi di gateway SiteWise Edge Anda dengan menambahkan kebijakan IAM berikut ke peran IAM gateway SiteWise Edge, ganti *<root-asset-id>* dengan ID aset root yang ingin Anda kirim ke gateway Edge. SiteWise

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringNotLike": {
          "iotsitewise:assetHierarchyPath": "/<root-asset-id>*"
        }
      }
    }
  ]
}
```

Jika ada aset saat ini di gateway SiteWise Edge Anda yang ingin Anda hapus, masuk ke gateway SiteWise Edge Anda dan jalankan perintah berikut untuk memaksa gateway SiteWise Edge disinkronkan AWS IoT SiteWise dengan menghapus cache.

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/  
sync_resource_bundles/edge.json
```

Konfigurasi dukungan proxy dan kelola toko kepercayaan untuk AWS IoT SiteWise Edge

Di AWS IoT SiteWise Edge, konfigurasi dan kelola toko kepercayaan untuk menyiapkan dukungan proxy untuk perangkat edge Anda. Pertama, atur konfigurasi proxy, lalu konfigurasi toko kepercayaan. Anda dapat mengonfigurasi penyimpanan kepercayaan baik selama instalasi gateway atau secara manual setelah gateway Anda dibuat.

- Proxy — Memfasilitasi konektivitas antara perangkat dan AWS layanan edge Anda di berbagai lingkungan jaringan.
- Trust store — Pastikan koneksi aman dengan mengelola sertifikat tepercaya. Konfigurasi yang tepat membantu Anda mematuhi kebijakan keamanan jaringan, mengaktifkan komunikasi di lingkungan jaringan terbatas, dan mengoptimalkan transfer data antara perangkat edge dan layanan cloud.

SiteWise Edge menggunakan beberapa toko kepercayaan untuk berbagai jenis komponen, memastikan aliran data yang aman dan efisien dari perangkat edge Anda ke cloud. Anda dapat mengonfigurasi toko kepercayaan dan proxy pada gateway yang ada atau selama proses instalasi saat membuat gateway baru.

Persyaratan untuk penyimpanan kepercayaan dan konfigurasi proxy

Sebelum Anda mengonfigurasi toko kepercayaan atau menginstal SiteWise Edge dengan pengaturan proxy, pastikan Anda memenuhi prasyarat. Ada beragam persyaratan implementasi berdasarkan penggunaan komponen dan persyaratan fungsionalitas Anda.

Persyaratan dukungan proxy

- URL server proxy Anda. URL harus menyertakan info pengguna, nomor port untuk host. Misalnya, `scheme://[userinfo@]host[:port]`.

- `scheme`— Harus HTTP atau HTTPS
- (Opsional) `userinfo` — Informasi nama pengguna dan kata sandi
- `host`— Nama host atau alamat IP dari server proxy
- `port`— Nomor port
- Daftar alamat untuk mem-bypass proxy.
- (Opsional) File sertifikat CA proxy jika Anda menggunakan proxy HTTPS dengan sertifikat yang ditandatangani sendiri.

Persyaratan toko kepercayaan

- Untuk fungsionalitas paket pemrosesan data lengkap dengan proxy HTTPS, Anda harus memperbarui ketiga toko kepercayaan.
- Jika Anda hanya menggunakan kolektor IoT SiteWise OPC UA dan SiteWise penerbit IoT, perbarui sertifikat AWS IoT Greengrass Core dan Java trust store ke versi terbaru.

Praktik terbaik untuk penyimpanan kepercayaan dan konfigurasi tepi server proxy

Untuk pemeliharaan berkelanjutan dan untuk mempertahankan tingkat keamanan tertinggi di lingkungan tepi Anda:

- Tinjau dan perbarui pengaturan proxy secara teratur agar sesuai dengan persyaratan keamanan jaringan Anda.
- Pantau konektivitas gateway dan aliran data untuk memastikan komunikasi proxy yang tepat
- Menjaga dan memperbarui toko kepercayaan sesuai dengan kebijakan manajemen sertifikat organisasi Anda
- Anda dapat menerapkan dan mengikuti praktik terbaik yang kami rekomendasikan untuk komunikasi yang aman di lingkungan tepi, seperti:
- Dokumentasikan konfigurasi proxy dan trust store Anda untuk visibilitas operasional
- Ikuti praktik keamanan organisasi Anda untuk manajemen kredensial

Praktik ini membantu menjaga operasi yang aman dan andal untuk gateway SiteWise Edge Anda sambil tetap selaras dengan kebijakan keamanan Anda yang lebih luas.

Konfigurasi pengaturan proxy selama instalasi gateway AWS IoT SiteWise Edge

Anda dapat mengonfigurasi AWS IoT SiteWise Edge untuk bekerja dengan server proxy selama instalasi gateway. Skrip instalasi mendukung proxy HTTP dan HTTPS dan dapat secara otomatis mengonfigurasi penyimpanan kepercayaan untuk koneksi proxy yang aman.

Ketika Anda menjalankan skrip instalasi dengan pengaturan proxy, ia melakukan beberapa tugas penting:

- Memvalidasi format dan parameter URL proxy untuk memastikannya ditentukan dengan benar.
- Mengunduh dan menginstal dependensi yang diperlukan melalui proxy yang dikonfigurasi.
- Jika sertifikat CA proxy disediakan, itu ditambahkan ke sertifikat CA AWS IoT Greengrass root dan diimpor ke Java KeyStore.
- Mengonfigurasi AWS IoT Greengrass (yang digunakan SiteWise Edge) untuk menggunakan proxy untuk semua koneksi keluar.
- Menyelesaikan instalasi SiteWise Edge dengan konfigurasi proxy dan trust store yang sesuai.

Untuk mengonfigurasi pengaturan proxy saat menginstal perangkat lunak gateway

1. Buat gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Buat gateway SiteWise Edge yang dihosting sendiri](#) dan [Instal perangkat lunak gateway AWS IoT SiteWise Edge di perangkat lokal Anda](#).
2. Jalankan skrip instalasi dengan pengaturan proxy yang sesuai untuk lingkungan Anda. Ganti placeholder dengan informasi proxy spesifik Anda

Ganti setiap item berikut:

- `-p, --proxy-url` — URL dari server proxy. URL harus salah satu `http` atau `https`.
- `-n, --no-proxy` — Daftar alamat yang dipisahkan koma untuk melewati proxy.
- (Opsional) `-c, --proxy-ca-cert` — Jalur ke file sertifikat CA proxy.
- (Opsional) `-j, --javastorepass` — KeyStore Kata sandi Java. Kata sandi defaultnya adalah `changeit`.

Linux

Untuk sistem Linux, gunakan struktur perintah berikut:

```
sudo ./install.sh -p proxy-url -n no-proxy-addresses [-c proxy-ca-cert-path] [-j javastorepass]
```

Windows

Untuk Microsoft Windows sistem yang menggunakan PowerShell, gunakan struktur perintah ini:

```
.\install.ps1 -ProxyUrl proxy-url -NoProxyAddresses no-proxy-addresses [-ProxyCaCertPath proxy-ca-cert-path] [-JavaStorePass javastorepass]
```

Pemecahan masalah selama instalasi berkemampuan proxy

Untuk informasi selengkapnya tentang menyelesaikan masalah trust store yang terkait dengan gateway SiteWise Edge, lihat [Masalah instalasi berkemampuan proxy](#).

Konfigurasi penyimpanan kepercayaan secara manual untuk dukungan proxy HTTPS di AWS IoT SiteWise Edge

Saat mengonfigurasi komponen AWS IoT SiteWise Edge untuk terhubung melalui proxy HTTPS, tambahkan sertifikat server proxy ke toko kepercayaan yang sesuai. SiteWise Edge menggunakan beberapa toko kepercayaan untuk mengamankan komunikasi. Ada tiga toko kepercayaan dan penggunaan Anda bergantung pada jenis komponen SiteWise Edge dalam implementasi gateway Anda.

Toko kepercayaan diperbarui secara otomatis selama proses instalasi ketika pengaturan proxy disediakan.

- [Konfigurasi penyimpanan kepercayaan komponen AWS IoT Greengrass Inti](#)— Sertifikat CA AWS IoT Greengrass root disertakan dalam toko kepercayaan untuk memverifikasi keaslian AWS layanan.

Toko kepercayaan ini membantu AWS IoT Greengrass komponen berkomunikasi dengan aman dengan AWS layanan melalui proxy sambil memverifikasi keaslian layanan tersebut.

- [Konfigurasi toko kepercayaan komponen berbasis Java](#)— Java KeyStore (JKS) adalah toko kepercayaan utama yang digunakan oleh komponen berbasis Java untuk koneksi. SSL/TLS

Aplikasi Java mengandalkan JKS untuk membangun koneksi yang aman. Misalnya, jika Anda menggunakan SiteWise penerbit IoT atau kolektor IoT SiteWise OPC UA, yang berbasis Java, Anda harus mengonfigurasi toko kepercayaan ini. Ini memastikan komponen ini dapat berkomunikasi dengan aman melalui proxy HTTPS saat mengirim data ke cloud atau mengumpulkan data dari server OPC UA.

- [Konfigurasi penyimpanan kepercayaan komponen tingkat sistem](#) Saat menggunakan proxy HTTPS, sertifikat mereka harus ditambahkan ke toko kepercayaan yang sesuai untuk mengaktifkan koneksi yang aman.

Saat menggunakan proxy HTTPS, sertifikat mereka harus ditambahkan ke toko kepercayaan yang sesuai untuk mengaktifkan koneksi yang aman. Ini diperlukan karena komponen tingkat sistem, sering ditulis dalam bahasa seperti Rust atau Go, bergantung pada penyimpanan kepercayaan sistem daripada JKS Java. Misalnya, jika Anda menggunakan utilitas sistem yang perlu berkomunikasi melalui proxy (seperti untuk pembaruan perangkat lunak atau sinkronisasi waktu), Anda harus mengonfigurasi penyimpanan kepercayaan tingkat sistem. Ini memastikan komponen dan utilitas ini dapat membangun koneksi aman melalui proxy.

Konfigurasi penyimpanan kepercayaan komponen AWS IoT Greengrass Inti

Untuk fungsi AWS IoT Greengrass Core yang menggunakan root CA Amazon:

1. Temukan file sertifikat di `/greengrass/v2/AmazonRootCA1.pem`
2. Tambahkan sertifikat root proxy HTTPS (ditandatangani sendiri) ke file ini.

```
-----BEGIN CERTIFICATE-----
MIIEFTCCAv2gAwIQWgIVAMHSAzWG/5YVRYtRQ0xXUTEpHuEmApzGCSqGSIb3DQEK
\nCwUAhuL9MQswCQwJVUzEPMAVUzEYMBYGA1UECgwP1hem9uLmNvbSBjbmuMRww
... content of proxy CA certificate ...
+vHIRlt0e5JAm5\noTIZGoFbK82A0/n07f/t5PSIDAim9V3Gc3pSXxCCAQoFYnui
GaPUlGk1gCE84a0X\n7Rp/1ND/PuMZ/s8Yj1kY2NmYmNjMCAXDTE5MTEyN2cM216
gJMIADggEPADf2/m45hzEXAMPLE=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDQTCCAimgF6AwIBAgITBmyfz/5mjAo54vB4ikPmljZKyjANJmApzyMZFo6qBg
ADA5MQswCQYDVQQGEwJVUzEPMA0tMVT8QtpHRh8jrdkGA1UEChMGDV3QQDExBBKW
... content of root CA certificate ...
```

```
o/ufQJQWUCyziar1hem9uMRkwFwYVPSHCb2XV4cdFyQzR1K1dZwgJcIQ6XUDgHaa
5MsI+yMRQ+hDaXJioblXgjUka642M4UwtBV8oK2xJNDd2ZhwLnoQdeXeGADKkpy
rqXRfKoQnoZsG4q5WTP46EXAMPLE
-----END CERTIFICATE-----
```

Konfigurasi proxy HTTPS pada gateway yang sudah mapan

Anda dapat menambahkan dukungan proxy ke gateway yang sudah mapan dengan menghubungkan ke port 443, bukan port 8883. Untuk informasi selengkapnya tentang menggunakan server proxy, lihat [Connect on port 443 atau melalui proxy jaringan](#) di Panduan AWS IoT Greengrass Version 2 Pengembang. Jika Anda membuat gateway baru, Anda dapat mengatur konfigurasi proxy selama instalasi gateway. Untuk informasi selengkapnya, lihat [Konfigurasi pengaturan proxy selama instalasi gateway AWS IoT SiteWise Edge](#).

Saat Anda menggunakan proxy HTTPS dengan AWS IoT Greengrass on SiteWise Edge, perangkat lunak secara otomatis memilih antara HTTP dan HTTPS untuk koneksi proxy berdasarkan URL yang disediakan.

 Important

Perbarui semua toko kepercayaan yang diperlukan sebelum mencoba terhubung melalui proxy HTTPS.

Konfigurasi toko kepercayaan komponen berbasis Java

Untuk SiteWise penerbit IoT, pengumpul IoT SiteWise OPC UA, dan layanan Java dalam paket pemrosesan data, lokasi penyimpanan kepercayaan Java default adalah `$JAVA_HOME/jre/lib/security/cacerts`

Untuk menambahkan sertifikat

1. Buat file untuk menyimpan sertifikat server proxy, seperti `proxy.crt`.

 Note

Buat file sebelumnya menggunakan sertifikat server proxy.

2. Tambahkan file ke toko kepercayaan Java menggunakan perintah berikut:

```
sudo keytool -import -alias proxyCert -keystore /usr/lib/jvm/java-11-openjdk-amd64/lib/security/cacerts -file proxy.crt
```

3. Saat diminta, gunakan kata sandi default: changeit

Konfigurasi penyimpanan kepercayaan komponen tingkat sistem

Untuk komponen yang ditulis dalam Rust, Go, dan bahasa lain yang menggunakan penyimpanan kepercayaan sistem:

Linux

Sistem Linux: Tambahkan sertifikat ke `/etc/ssl/certs/ca-certificates.crt`

Windows

Microsoft Windows sistem: Untuk mengonfigurasi toko kepercayaan, ikuti prosedur [Certificate Store](#) di dokumentasi Microsoft Ignite.

Windows menawarkan beberapa toko sertifikat, termasuk toko terpisah untuk cakupan Pengguna dan Komputer, masing-masing dengan beberapa sub-toko. Untuk sebagian besar pengaturan SiteWise Edge, kami sarankan untuk menambahkan sertifikat ke toko. `COMPUTER | Trusted Root Certification Authorities` Namun, tergantung pada konfigurasi spesifik dan persyaratan keamanan Anda, Anda mungkin perlu menggunakan toko yang berbeda.

Memecahkan masalah toko kepercayaan

Untuk informasi selengkapnya tentang menyelesaikan masalah trust store yang terkait dengan gateway SiteWise Edge, lihat [Masalah toko kepercayaan](#).

Gunakan AWS IoT SiteWise APIs di tepi

AWS IoT SiteWise menyediakan subset dari APIs, bersama dengan spesifik tepi APIs, memungkinkan interaksi tanpa batas dengan model aset dan aset terkait yang digunakan di edge. Model aset ini harus dikonfigurasi untuk berjalan di tepi. Untuk informasi selengkapnya, lihat [Konfigurasi model aset untuk pemrosesan data di SiteWise Edge](#) petunjuk terperinci tentang proses penyiapan ini.

Setelah mengonfigurasinya APIs, Anda dapat mengambil data komprehensif tentang model aset dan aset individual Anda. Mengambil model aset, aset, dasbor, portal, dan informasi proyek dapat

membantu Anda memantau portal dan dasbor yang digunakan, serta mengakses data aset yang dikumpulkan di tingkat tepi. Ini menyediakan host pusat di jaringan Anda untuk interaksi dengan AWS IoT SiteWise tanpa memerlukan panggilan API web.

Topik

- [Semua perangkat AWS IoT SiteWise Edge yang tersedia APIs](#)
- [Hanya tepi APIs untuk digunakan dengan AWS IoT SiteWise perangkat tepi](#)
- [Aktifkan CORS di Edge AWS IoT SiteWise APIs](#)
- [Konfigurasi batas waktu sesi untuk Edge AWS IoT SiteWise](#)
- [Tutorial: Daftar model aset pada gateway AWS IoT SiteWise Edge](#)

Semua perangkat AWS IoT SiteWise Edge yang tersedia APIs

AWS IoT SiteWise menyediakan berbagai APIs untuk digunakan pada perangkat edge sehingga Anda dapat menyelesaikan tugas secara lokal di perangkat. Beberapa edge yang tersedia APIs termasuk mengambil model aset, membuat dan memperbarui properti aset, dan mengirim aliran data ke cloud. Dengan memanfaatkan ini APIs, Anda dapat membangun solusi yang dapat beroperasi di lingkungan dengan konektivitas jaringan intermiten atau terbatas.

Tersedia AWS IoT SiteWise APIs

Berikut ini AWS IoT SiteWise APIs tersedia di perangkat edge:

- [ListAssetModels](#)
- [DescribeAssetModel](#)
- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)
- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)

- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

Tersedia hanya tepi APIs

Berikut ini APIs digunakan secara lokal pada perangkat di tepi:

- [Mengautentikasi](#)— Gunakan API ini untuk mendapatkan kredensial sementara SigV4 yang akan Anda gunakan untuk melakukan panggilan API.

Hanya tepi APIs untuk digunakan dengan AWS IoT SiteWise perangkat tepi

Selain AWS IoT SiteWise APIs yang tersedia di tepi, ada yang khusus tepi. Tepi-specific tersebut dijelaskan di bawah APIs ini.

Mengautentikasi

Mendapat kredensial dari gateway SiteWise Edge. Anda harus menambahkan pengguna lokal atau terhubung ke sistem Anda menggunakan LDAP atau kumpulan pengguna Linux. Untuk informasi selengkapnya tentang menambahkan pengguna, lihat [kumpulan pengguna LDAP atau Linux](#).

Sintaksis Permintaan

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
  "username": "string",
  "password": "string",
  "authMechanism": "string"
}
```

Parameter permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi permintaan

Permintaan menerima data berikut dalam format JSON.

nama pengguna

Nama pengguna yang digunakan untuk memvalidasi panggilan permintaan.

Tipe: String

Diperlukan: Ya

password

Kata sandi pengguna yang meminta kredensial.

Tipe: String

Diperlukan: Ya

AuthMechanism

Metode otentikasi untuk memvalidasi pengguna ini di host.

Tipe: String

Nilai valid: `ldap`, `linux`, `winnt`

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json
{
  "accessKeyId": "string",
  "secretAccessKey": "string",
  "sessionToken": "string",
  "region": "edge"
}
```

Elemen jawaban

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Data berikut dikembalikan dalam format JSON.

accessKeyId

ID kunci akses yang mengidentifikasi kredensi keamanan sementara.

Kendala Panjang: Panjang minimum 16. Panjang maksimum 128.

Pola: `[\w]*`

secretAccessKey

Kunci akses rahasia yang dapat digunakan untuk menandatangani permintaan.

Tipe: String

sessionToken

Token yang harus diteruskan pengguna ke API layanan untuk menggunakan kredensial sementara.

Tipe: String

region

Wilayah yang Anda targetkan untuk panggilan API.

Jenis: KONSTAN - edge

Kesalahan

IllegalArgumentException

Permintaan ditolak karena dokumen badan yang disediakan cacat. Pesan kesalahan menjelaskan kesalahan tertentu.

Kode Status HTTP: 400

AccessDeniedException

Pengguna tidak memiliki kredensial yang valid berdasarkan Penyedia Identitas saat ini. Pesan kesalahan menjelaskan Mekanisme otentikasi.

Kode Status HTTP: 403

TooManyRequestsException

Permintaan telah mencapai batas upaya otentikasi. Pesan kesalahan berisi jumlah waktu untuk menunggu sampai upaya otentikasi baru dilakukan.

Kode Status HTTP: 429

Aktifkan CORS di Edge AWS IoT SiteWise APIs

Mengaktifkan CORS (Cross-Origin Resource Sharing) di AWS IoT SiteWise Edge APIs memungkinkan aplikasi web untuk berkomunikasi langsung dengan APIs berbagai domain. Hal ini memungkinkan integrasi tanpa batas, pertukaran data real-time, dan akses data lintas domain tanpa server perantara atau solusi. Pengaturan CORS dapat dikonfigurasi untuk menentukan asal yang diizinkan, memastikan akses lintas asal yang terkontrol.

Note

CORS tersedia untuk versi 3.3.1 dan yang lebih baru dari Fitur ini tersedia untuk versi 3.3.1 dan yang lebih baru dari komponen `aws.iot.SiteWiseEdgeProcessor`. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise prosesor](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Untuk mengaktifkan CORS di SiteWise Edge APIs

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway SiteWise Edge yang ingin Anda aktifkan CORS. Anda dapat mengaktifkan CORS pada jenis AWS IoT Greengrass V2 penerapan.
4. Di bagian konfigurasi Gateway, pilih perangkat inti Greengrass terkait.
5. Di tab Deployment, di bawah Perangkat Greengrass, pilih tautan penerapan yang sesuai.
6. Di bawah Tindakan pilih Revisi, lalu Revisi penerapan.

Important

Membuat konfigurasi berkemampuan CORS yang direvisi menggantikan konfigurasi perangkat saat ini.

7. Di Langkah 1, Tentukan target, berikan Nama opsional untuk mengidentifikasi penyebaran.
8. Pada Langkah 2, Pilih komponen - opsional, Anda dapat membiarkan semua pilihan saat ini apa adanya dan memilih Berikutnya.

9. Pada Langkah 3, Konfigurasikan komponen - opsional, pilih `aws.iot.SiteWiseEdgeProcessor`, dan pilih `Configure component`.
10. Di bagian Pembaruan konfigurasi, di bawah Konfigurasi untuk digabungkan, masukkan JSON berikut:

```
{
  "AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN": "*"
}
```

 Note

Menggunakan `*` sebagai nilai untuk `AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` memungkinkan semua asal. Untuk lingkungan produksi, disarankan untuk menentukan asal yang tepat URLs untuk keamanan yang lebih baik.

11. Pilih `Konfirmasi`.
12. Pilih Berikutnya untuk melanjutkan melalui langkah-langkah yang tersisa sampai Anda tiba di `Step5, Review`.
13. Tinjau perubahan konfigurasi Anda, lalu pilih `Deploy` untuk menerapkan perubahan ke gateway `SiteWise Edge` Anda.

 Note

Atau, Anda dapat mengaktifkan CORS dengan menyetel global variabel lingkungan `AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` ke `*` AWS IoT SiteWise gateway Anda.

 Note

Untuk proxy yang diautentikasi, `userinfo` harus disertakan dalam `url` bidang dalam konfigurasi proxy daripada sebagai terpisah `username` dan `password` bidang.

Setelah penerapan selesai, CORS diaktifkan di `SiteWise Edge API` Anda, memungkinkan asal yang ditentukan untuk membuat permintaan lintas asal ke API.

Konfigurasi batas waktu sesi untuk Edge AWS IoT SiteWise

SiteWise Edge memungkinkan Anda mengonfigurasi batas waktu sesi untuk SiteWise Edge API. Fitur ini meningkatkan keamanan dengan secara otomatis menghentikan sesi tidak aktif setelah jangka waktu tertentu. Bagian ini memandu Anda melalui proses mengonfigurasi batas waktu sesi menggunakan Konsol AWS IoT SiteWise

Note

Konfigurasi batas waktu sesi tersedia untuk versi 3.4.0 dan yang lebih baru dari komponen `aws.iot.SiteWiseEdgeProcessor`. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise prosesor](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Untuk mengonfigurasi batas waktu sesi untuk gateway SiteWise Edge

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway SiteWise Edge tempat Anda ingin mengonfigurasi batas waktu sesi.

Note

Anda dapat mengonfigurasi batas waktu sesi pada jenis AWS IoT Greengrass V2 penerapan.

4. Di bagian konfigurasi Gateway, pilih perangkat inti Greengrass terkait.
5. Di tab Deployment, di bawah Perangkat Greengrass, pilih tautan penerapan yang sesuai.
6. Di bawah Tindakan pilih Revisi. Baca peringatan, lalu pilih Revise deployment.

Important

Membuat konfigurasi batas waktu sesi yang direvisi menggantikan konfigurasi perangkat saat ini.

7. Pada Langkah 1, Tentukan target, berikan Nama opsional untuk mengidentifikasi penyebaran yang direvisi, lalu pilih Berikutnya.
8. Pada Langkah 2, Pilih komponen - opsional, Anda dapat membiarkan semua pilihan saat ini apa adanya dan memilih Berikutnya.

9. Pada Langkah 3, Konfigurasi komponen - opsional, pilih `aws.iot.SiteWiseEdgeProcessor`, dan pilih `Configure component`.
10. Di bagian Pembaruan konfigurasi, di bawah Konfigurasi untuk digabungkan, masukkan JSON berikut:

```
{
  "AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES": "240"
}
```

11. Tetapkan nilainya `AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES` dalam hitungan menit. Nilai batas waktu sesi dapat dari 1 menit hingga 10080 menit (7 hari). Nilai default adalah 240 menit (4 jam).
12. Pilih `Konfirmasi`.
13. Pilih `Berikutnya` untuk melanjutkan melalui langkah-langkah yang tersisa sampai Anda tiba di Langkah 5, `Tinjau`.
14. Tinjau perubahan konfigurasi Anda, lalu pilih `Deploy` untuk menerapkan perubahan ke gateway `SiteWise Edge` Anda.

Note

Atau, Anda dapat mengonfigurasi batas waktu sesi dengan menyetel variabel lingkungan global `AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES` ke nilai yang Anda inginkan (dalam hitungan menit) di gateway `Edge` Anda. `SiteWise`

Setelah penerapan selesai, konfigurasi batas waktu sesi baru diterapkan ke `SiteWise Edge` API Anda.

Tutorial: Daftar model aset pada gateway AWS IoT SiteWise Edge

Anda dapat menggunakan subset yang tersedia AWS IoT SiteWise APIs bersama dengan `edge-specific` APIs untuk berinteraksi dengan model aset dan asetnya di `edge`. Tutorial ini akan memandu Anda untuk mendapatkan kredensial sementara ke gateway AWS IoT SiteWise Edge dan mendapatkan daftar model aset di gateway `SiteWise Edge`.

Prasyarat

Dalam langkah-langkah tutorial ini Anda dapat menggunakan berbagai alat. Untuk menggunakan alat ini, pastikan Anda telah menginstal prasyarat yang sesuai.

Untuk menyelesaikan tutorial ini, Anda memerlukan hal berikut:

- Sebuah komputer yang diarahkan dan berjalan [AWS IoT SiteWise Persyaratan gateway yang dihosting sendiri di Edge](#)
- Akses ke gateway SiteWise Edge Anda di jaringan yang sama melalui port 443.
- [OpenSSL diinstal](#)
- (AWS OpsHub untuk AWS IoT SiteWise) [AWS IoT SiteWise Aplikasi AWS OpsHub untuk](#)
- (curl) [curl](#) terpasang
- [\(Python\) urllib3 diinstal](#)
- [\(Python\) Python3 diinstal](#)
- [\(Python\) Boto3 diinstal](#)
- (Python) diinstal [BotoCore](#)

Langkah 1: Dapatkan sertifikat yang ditandatangani layanan gateway SiteWise Edge

Untuk membuat koneksi TLS ke yang APIs tersedia di gateway SiteWise Edge, Anda memerlukan sertifikat tepercaya. Anda dapat membuat sertifikat ini menggunakan AWS OpsHub OpenSSL atau for. AWS IoT SiteWise

OpenSSL

Note

Anda perlu [OpenSSL](#) diinstal untuk menjalankan perintah ini.

Buka terminal dan jalankan perintah berikut untuk mendapatkan sertifikat yang ditandatangani dari gateway SiteWise Edge. Ganti `<sitewise_gateway_ip>` dengan IP gateway SiteWise Edge.

```
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

Anda dapat menggunakan AWS OpsHub untuk AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Kelola SiteWise gateway Edge](#).

Jalur absolut ke sertifikat gateway SiteWise Edge yang diunduh digunakan dalam tutorial ini. Jalankan perintah berikut untuk mengekspor jalur lengkap sertifikat Anda, ganti `<absolute_path_to_certificate>` dengan jalur ke sertifikat:

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

Langkah 2: Dapatkan nama host gateway SiteWise Edge Anda

Note

Anda perlu [OpenSSL](#) diinstal untuk menjalankan perintah ini.

Untuk menyelesaikan tutorial, Anda memerlukan nama host gateway SiteWise Edge Anda. Untuk mendapatkan nama host gateway SiteWise Edge Anda, jalankan yang berikut ini, ganti `<sitewise_gateway_ip>` dengan IP gateway SiteWise Edge:

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po  
'CN = \K.*' | head -1
```

Jalankan perintah berikut untuk mengekspor nama host untuk digunakan nanti, ganti `<your_edge_gateway_hostname>` dengan nama host gateway SiteWise Edge Anda:

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

Langkah 3: Dapatkan kredensial sementara untuk gateway Edge Anda SiteWise

Sekarang setelah Anda memiliki sertifikat yang ditandatangani dan nama host gateway SiteWise Edge Anda, Anda perlu mendapatkan kredensial sementara sehingga Anda dapat berjalan APIs di gateway. Anda bisa mendapatkan kredensial ini melalui AWS OpsHub untuk AWS IoT SiteWise atau langsung dari gateway SiteWise Edge menggunakan APIs

 Important

Kredensyal kedaluwarsa setiap 4 jam, jadi Anda harus mendapatkan kredensialnya sebelum menggunakan gateway di APIs Edge Anda. SiteWise Jangan menyimpan kredensil selama lebih dari 4 jam.

Dapatkan kredensyal sementara menggunakan for AWS OpsHubAWS IoT SiteWise

 Note

Anda memerlukan [AWS IoT SiteWise aplikasi AWS OpsHub for](#) diinstal.

Untuk menggunakan AWS IoT SiteWise aplikasi AWS OpsHub untuk mendapatkan kredensyal sementara Anda lakukan hal berikut:

1. Masuk ke aplikasi.
2. Pilih Pengaturan.
3. Untuk Otentikasi, pilih Salin kredensyal.
4. Perluas opsi yang sesuai dengan lingkungan Anda dan pilih Salin.
5. Simpan kredensialnya untuk digunakan nanti.

Dapatkan kredensyal sementara menggunakan SiteWise Edge gateway API

Untuk menggunakan SiteWise Edge gateway API untuk mendapatkan kredensyal sementara, Anda dapat menggunakan skrip Python atau curl, pertama-tama Anda harus memiliki nama pengguna dan kata sandi untuk gateway Edge Anda. SiteWise Gateway SiteWise Edge menggunakan otentikasi dan otorisasi SiGv4. Untuk informasi selengkapnya tentang menambahkan pengguna, lihat [kumpulan pengguna LDAP atau Linux](#). Kredensyal ini akan digunakan dalam langkah-langkah berikut untuk mendapatkan kredensyal lokal di gateway SiteWise Edge Anda yang diperlukan untuk menggunakan. AWS IoT SiteWise APIs

Python

Note

[Anda perlu menginstal urllib3 dan Python3.](#)

Untuk mendapatkan kredensialnya menggunakan Python

1. Buat file bernama `get_credentials.py` dan salin kode berikut ke dalamnya.

```
'''
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. Jalankan `get_credentials.py` dari terminal menggantikan `<gateway_username>` dan `<gateway_password>` dengan kredensial yang Anda buat.

```
python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE  
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'
```

curl

 Note

Anda perlu menginstal [curl](#).

Untuk mendapatkan kredensial menggunakan curl

1. Jalankan perintah berikut dari penggantian terminal `<gateway_username>` dan `<gateway_password>` dengan kredensial yang Anda buat.

```
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"  
}'
```

Responsnya akan terlihat seperti berikut:

```
{  
    "username": "sweuser",  
    "accessKeyId": "<accessKeyId>",
```

```
"secretAccessKey": "<secretAccessKey>",
"sessionToken": "<sessionToken>",
"sessionExpiryTime": "2022-11-17T04:51:40.927095Z",
"authMechanism": "linux",
"role": "edge-user"
}
```

2. Jalankan perintah berikut dari terminal Anda.

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

Langkah 4: Dapatkan daftar model aset di gateway SiteWise Edge

Sekarang setelah Anda memiliki sertifikat yang ditandatangani, nama host gateway SiteWise Edge Anda, dan kredensial sementara untuk gateway SiteWise Edge Anda, Anda dapat menggunakan ListAssetModels API untuk mendapatkan daftar model aset di gateway Edge Anda SiteWise .

Python

Note

Anda perlu [Python3](#), [Boto3](#), dan diinstal. [BotoCore](#)

Untuk mendapatkan daftar model aset menggunakan Python

1. Buat file bernama list_asset_model.py dan salin kode berikut ke dalamnya.

```
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. Jalankan `list_asset_model.py` dari terminal.

```
python3 list_asset_model.py
```

curl

Note

Anda perlu menginstal [curl](#).

Untuk mendapatkan daftar model aset menggunakan curl

Jalankan perintah berikut dari terminal.

```
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"
```

Responsnya akan terlihat seperti berikut:

```
{
  "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  
}
```

Host gateway SiteWise Edge di Siemens Industrial Edge

Host gateway Anda di Siemens Industrial Edge menggunakan aplikasi AWS IoT SiteWise Edge. Sama seperti halnya AWS IoT Greengrass V2, Anda dapat mengoptimalkan proses manufaktur atau meningkatkan alur kerja operasional menggunakan SiteWise Edge on Siemens Industrial Edge.

Anda dapat menyerap data dari perangkat Siemens Industrial Edge ke AWS akun Anda dengan menjalankan gateway SiteWise Edge di perangkat. Untuk melakukan ini, minta akses ke aplikasi AWS IoT SiteWise Edge dari tim dukungan SiteWise Edge. Kemudian, buat sumber daya gateway SiteWise Edge dengan target penerapan perangkat Siemens Industrial Edge - baru. Selanjutnya, unduh file konfigurasi, dan unggah ke aplikasi Anda melalui Siemens Industrial Edge Management portal. Untuk informasi selengkapnya tentang menjalankan aplikasi Siemens Industrial Edge, termasuk cara menyiapkan Siemens sumber daya yang diperlukan, lihat [Apa itu Industrial Edge?](#) dalam dokumentasi Siemens.

Note

Siemens bukan vendor atau pemasok untuk SiteWise Edge. Siemens Industrial EdgeMarketplace adalah pasar independen.

Topik

- [Keamanan](#)
- [Siemens Secure Storage dan aplikasi AWS IoT SiteWise Edge](#)

- [Tujuan untuk Siemens Industrial Edge perangkat](#)
- [Migrasi dari aplikasi pratinjau](#)
- [Pemecahan Masalah](#)
- [AWS IoT SiteWise Changelog aplikasi tepi](#)
- [Persyaratan untuk aplikasi AWS IoT SiteWise Edge](#)
- [Buat gateway untuk Siemens Industrial Edge](#)
- [Buat Siemens Databus user untuk aplikasi](#)
- [Akses aplikasi AWS IoT SiteWise Edge](#)
- [Instal aplikasi ke Siemens perangkat](#)
- [Perbarui konfigurasi aplikasi AWS IoT SiteWise Edge](#)

Keamanan

Sebagai bagian dari [Model Tanggung Jawab Bersama](#) antara AWS, pelanggan kami, dan mitra kami, berikut ini menjelaskan siapa yang bertanggung jawab atas berbagai aspek keamanan:

Tanggung jawab pelanggan

- Memeriksa pasangan.
- Mengkonfigurasi akses jaringan yang diberikan kepada mitra.
- Secara fisik mengamankan perangkat yang menjalankan SiteWise Edge.

AWS tanggung jawab

- Mengisolasi mitra dari sumber daya AWS Cloud pelanggan.

Tanggung jawab mitra

- Menggunakan default aman.
- Menjaga solusi tetap aman dari waktu ke waktu melalui tambalan dan pembaruan lain yang sesuai.
- Menjaga kerahasiaan data pelanggan.
- Memeriksa aplikasi lain yang tersedia di pasar mitra.

Siemens Secure Storage dan aplikasi AWS IoT SiteWise Edge

Untuk melindungi kredensi dan rahasia yang diperlukan untuk menjalankan aplikasi AWS IoT SiteWise Edge, Siemens Industrial Edge sediakan mekanisme untuk menyimpan kredensial di

perangkat dengan aman. Aplikasi AWS IoT SiteWise Edge tidak akan berjalan di perangkat jika tidak memiliki dukungan untuk menyimpan kredensial ini dengan aman. Kegagalan jalankan yang disebabkan oleh dukungan Penyimpanan Aman yang hilang dicatat dalam file log.

Versi OS minimum berikut diperlukan untuk menginstal dan menjalankan aplikasi AWS IoT SiteWise Edge. Tingkatkan perangkat Anda ke versi terbaru untuk menginstal aplikasi.

- Untuk perangkat virtual: IEVD versi 1.19 atau lebih tinggi
- Untuk perangkat fisik: IED-OS versi 2.2 atau lebih tinggi

Aplikasi AWS IoT SiteWise Edge aktif tidak Siemens Industrial Edge akan berjalan sampai Anda memutakhirkan perangkat Anda.

Tujuan untuk Siemens Industrial Edge perangkat

Saat menggunakan aplikasi AWS IoT SiteWise Edge aktifSiemens Industrial Edge, tujuan membantu menyiapkan data sebelum mengirimnya AWS IoT SiteWise untuk analisis dan distribusi lebih lanjut. Anda dapat mengonfigurasi setelan tujuan data untuk konsumsi data buffer menggunakan Amazon S3 atau menggunakan konsumsi data waktu nyata. Keduanya memungkinkan Anda berlangganan topik MQTT menggunakan filter jalur pada target penyebaran Siemens Industrial Edge perangkat.

Target Siemens Industrial Edge penerapan pada gateway Anda mendukung dua metode penanganan data utama:

- AWS IoT SiteWise pengaturan waktu nyata - Data dikirim langsung ke AWS IoT SiteWise saat dikumpulkan
- AWS IoT SiteWise buffer menggunakan pengaturan Amazon S3 - Data dikumpulkan dan disimpan sementara dalam batch sebelum dikirim ke Amazon S3

Untuk informasi selengkapnya tentang mengonfigurasi opsi ini, lihat [Tambahkan tujuan AWS IoT SiteWise buffer menggunakan Amazon S3](#) dan [Tambahkan tujuan real-time AWS IoT SiteWise Edge](#).

Awalan untuk filter jalur

Filter jalur untuk gateway yang menggunakan target Siemens Industrial Edge penerapan menggabungkan topik dan nama aliran data untuk membuat pengenalan unik untuk data Anda. Topik gabungan dengan nama aliran data disebut awalan di Siemens Industrial Edge gateway. Ini berbeda dari gateway yang dihosting sendiri di mana filter jalur hanya didasarkan pada topik MQTT.

Example Struktur filter jalur untuk aliran data Siemens

Filter jalur khas untuk aliran data Siemens mencakup jalur topik dan nama aliran data:

```
ie/d/device1/application1/datastream1
```

Di mana:

- `ie/d/` adalah awalan yang diperlukan untuk aliran data Siemens
- `device1/application1` mewakili jalur hierarkis
- `datastream1` adalah nama aliran data spesifik

Note

Saat bekerja dengan aliran Siemens Industrial Edge data, pastikan Anda menyertakan topik metadata (`ie/m/`) dan data (`ie/d/`) di filter jalur Anda untuk menerima informasi lengkap tentang aliran data Anda.

Tujuan dan filter jalur

Lihat topik berikut untuk mempelajari lebih lanjut tentang tujuan dan filter jalur di gateway berkemampuan MQTT:

- [Memahami tujuan AWS IoT SiteWise Edge](#)
- [Tambahkan tujuan real-time AWS IoT SiteWise Edge](#)
- [Tambahkan tujuan AWS IoT SiteWise buffer menggunakan Amazon S3](#)
- [Memahami filter jalur untuk tujuan AWS IoT SiteWise Edge](#)
- [Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge](#)
- [Kelola tujuan AWS IoT SiteWise Edge](#)

Migrasi dari aplikasi pratinjau

Jika Anda menjalankan SiteWise Edge Siemens Industrial Edge selama fase pratinjau, Anda harus meningkatkan dari versi pratinjau, versi 1.0.1, ke versi terbaru. Lakukan hal berikut untuk bermigrasi:

1. Buat gateway SiteWise Edge baru. Untuk informasi selengkapnya, lihat [Buat gateway untuk Siemens Industrial Edge](#).
2. Buat yang baru Siemens Databus user untuk setiap gateway baru. Untuk informasi selengkapnya, lihat [Buat Siemens Databus user untuk aplikasi](#).
3. Copot pemasangan aplikasi gateway AWS IoT SiteWise Edge versi 1.0.1 di IED Anda.

Note

Bersiaplah untuk interupsi aliran data saat Anda mengkonfigurasi ulang AWS IoT SiteWise aset yang sebelumnya digunakan oleh versi pratinjau aplikasi Edge. AWS IoT SiteWise Sementara riwayat data dipertahankan, ada potensi kehilangan data saat Anda menginstal ulang gateway baru.

4. Hapus gateway SiteWise Edge yang Anda buat selama pratinjau di [Konsol AWS IoT SiteWise](#)
5. Instal aplikasi gateway AWS IoT SiteWise Edge di IED menggunakan file konfigurasi gateway baru. Untuk informasi selengkapnya, lihat [Instal aplikasi ke Siemens perangkat](#).

Important

Menginstal gateway baru menimpa versi pratinjau aplikasi SiteWise Edge. Tidak mungkin untuk kembali ke versi 1.0.1 setelah menginstal versi 2.0.0.

Setelah mengonfigurasi gateway baru dan Siemens Databus user, data Anda mengalir ke properti Anda.

Anda juga dapat memutakhirkan aplikasi SiteWise Edge Anda dari versi secara langsung. Namun, konfigurasi gateway baru masih diperlukan.

Pemecahan Masalah

Untuk memecahkan masalah gateway SiteWise Edge di Siemens Industrial Edge perangkat Anda, lihat [Memecahkan masalah aplikasi AWS IoT SiteWise Edge pada Siemens Industrial Edge](#)

Anda juga dapat mengakses [AWS re:Post](#) untuk menemukan jawaban atas pertanyaan Anda.

AWS IoT SiteWise Changelog aplikasi tepi

Tabel berikut menjelaskan perubahan di setiap versi aplikasi AWS IoT SiteWise Edge.

Versi	Perubahan	
3.0.0	Fitur baru • Menambahkan opsi konfigurasi tambahan di AWS IoT SiteWise konsol untuk gateway yang diinstal pada: Siemens Industrial Edge • Tambahkan dan konfigurasi tujuan Amazon S3. • Hapus tujuan AWS IoT SiteWise real-time. • Tambahkan filter untuk setiap tujuan untuk mengontrol perutean data. • Menambahkan opsi untuk konfigurasi awalan global untuk mengirim semua data dari penerbit SiteWise IoT di penginstal. Siemens IE Awalan diterapkan setelah penyaringan. Perbaikan bug dan peningkatan • Menambahkan kemampuan untuk mencoba lagi koneksi ke Siemens Industrial Edge Databus jika upaya awal aplikasi AWS IoT SiteWise Edge gagal.	
2.0.1	Perbaikan bug dan peningkatan • Memperbaiki masalah di mana aplikasi akan memasuki status kegagalan dan berhenti jika tidak dapat mengambil AWS kredensialnya saat startup. Menambahkan dukungan untuk mencoba lagi hingga pengambilan kredensial berhasil.	
2.0.0	• Aplikasi AWS IoT SiteWise Edge sekarang tersedia secara umum. • Aplikasi membutuhkan Siemens IEVD versi 1.19, atau Siemens IED-OS versi 2.2. • Peningkatan kinerja: Mengurangi penggunaan memori dan CPU. • Peningkatan debugging: Anda sekarang dapat mengunggah file konfigurasi opsional untuk mengaktifkan log debug.	

Versi	Perubahan	
	- Peningkatan keamanan: Aplikasi ini digunakan SecureStorage API untuk menyimpan kredensial dengan aman di perangkat. - Dockernilai intisari: sha256:4a960f29234a190ebb5224c1fd0f3e99faafccc4cb3d93ca13fef247b6656d18	
1.0.1	Rilis awal	

Persyaratan untuk aplikasi AWS IoT SiteWise Edge

Untuk menjalankan AWS IoT SiteWise Edge onSiemens Industrial Edge, Anda memerlukan yang berikut ini:

- Akun [Platform Pertukaran Digital Siemens](#).
- Akun Siemens Industrial Edge Hub (iehub).
- Sebuah Siemens Industrial Edge Management contoh.
 - Layanan Konfigurasi Aplikasi IE. Untuk mempelajari lebih lanjut, lihat [Installing the IE App Configuration Service manually](#)di Siemens Industrial Edge Managementdokumentasi.
- Akses ke versi 2.0.1 atau lebih tinggi dari aplikasi AWS IoT SiteWise Edge. Untuk informasi selengkapnya, lihat [Akses aplikasi AWS IoT SiteWise Edge](#).
- Baik Siemens Industrial Edge Perangkat (IED) atau Perangkat Siemens Industrial Edge virtual (IEVD).
 - Minimal ruang disk 15 GB untuk kebutuhan perangkat keras.
 - 1 GB RAM dengan tambahan 1 GB memori swap.
 - Konfigurasi perangkat untuk memungkinkan lalu lintas keluar pada port 443 dan 8883.
 - Prosesor x86-64 bit.
 - Siemens Industrial Edge Managementversi 1.13.10 atau lebih tinggi.
 - Kesesuaian perangkat dengan persyaratan. Siemens Secure Storage
 - Pada perangkat virtual, IEVD versi 1.19 atau lebih tinggi.
 - Pada perangkat fisik, IED-OS versi 2.2 atau lebih tinggi.
 - Versi terbaru dariDocker Compose.
 - Docker Engineversi 18.091 atau lebih tinggi.

- Akses domain yang diperlukan. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise titik akhir](#).

Buat gateway untuk Siemens Industrial Edge

Setelah Anda memiliki akun Siemens dan instans IEM yang tepat, Anda dapat membuat gateway SiteWise Edge dari perangkat jenis penyebaran Siemens Industrial Edge.

Note

Pastikan Anda memenuhi semua persyaratan untuk menjalankan perangkat aktif Siemens Industrial Edge Management. Untuk informasi selengkapnya, lihat [Persyaratan untuk aplikasi AWS IoT SiteWise Edge](#).

Untuk membuat file konfigurasi

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih Buat gateway.
4. Untuk jenis Deployment, pilih perangkat Siemens Industrial Edge.
5. Masukkan nama untuk gateway SiteWise Edge Anda atau gunakan nama yang dihasilkan oleh AWS IoT SiteWise.
6. (Opsional) Di bawah konfigurasi lanjutan, lakukan hal berikut:
 - Masukkan nama untuk AWS IoT Core Benda Anda atau gunakan nama yang dihasilkan oleh AWS IoT SiteWise.
7. Pilih Buat gateway.
8. Di kotak dialog Generate SiteWise Edge gateway configuration file, pilih Generate and download. AWS IoT SiteWise secara otomatis menghasilkan file konfigurasi yang akan Anda gunakan untuk mengkonfigurasi aplikasi AWS IoT SiteWise Edge.

Important

Anda menggunakan file konfigurasi gateway untuk membuat cadangan dan memulihkan aplikasi AWS IoT SiteWise Edge Anda. Simpan file konfigurasi gateway SiteWise Edge

Anda [AWS Secrets Manager](#) untuk menyimpan dan mengelola file dengan aman. Secrets Manager menyimpan, mengelola, dan mengambil informasi sensitif dengan aman.

Buat Siemens Databus user untuk aplikasi

AWS IoT SiteWise Edge on Siemens Industrial Edge mencerna data dari Siemens Databus aplikasi. Untuk menghubungkan SiteWise Edge ke Siemens Databus, Anda memerlukan Siemens Databus user yang menyediakan akses ke data yang ingin Anda transfer dengan aman. AWS IoT SiteWise Untuk memulai, buat Siemens Databus user dan kemudian berikan kredensialnya ke aplikasi SiteWise Edge.

Untuk membuat Siemens Databus user

1. Dalam Siemens Industrial Edge Management contoh Anda, pilih Edge Management di bagian Aplikasi Platform.
2. Pilih ikon Koneksi data.
3. Pilih Databus. Daftar perangkat Anda yang terhubung muncul.
4. Pilih perangkat untuk terhubung ke aplikasi AWS IoT SiteWise Edge.
5. Pilih Luncurkan. Databus Configurator Untuk perangkat yang Anda pilih muncul.
6. Buat pengguna untuk perangkat Edge Anda di bawah Pengguna. Untuk informasi selengkapnya tentang membuat pengguna, lihat [Pengguna](#) di dokumentasi Manajemen Tepi Industri Siemens.
7. Pilih topik yang Siemens Databus harus diakses. Topik-topik ini membatasi apa yang dapat diakses AWS IoT SiteWise Edge.

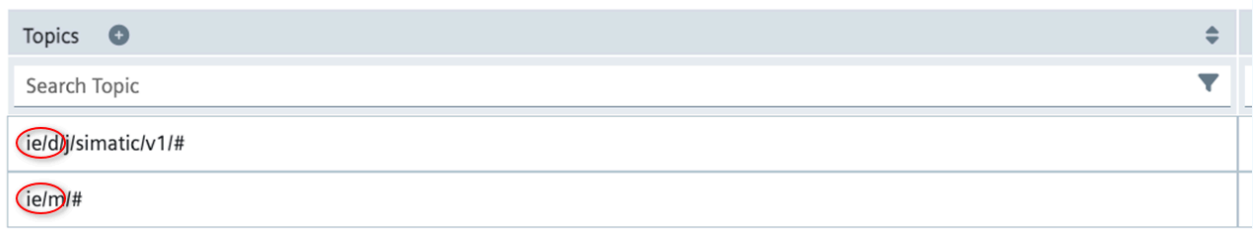
Important

Semua topik Siemens Databus user yang dapat diakses oleh a dipublikasikan AWS IoT SiteWise.

Note

Siemens Databus userS membutuhkan akses ke topik data dan metadata. Topik yang dimulai dengan ie/d adalah topik data. Dan topik yang dimulai dengan ie/m adalah

topik metadata. Bagikan topik secara berpasangan sehingga SiteWise Edge memiliki akses ke data dan metadata untuk setiap topik masing-masing.



8. Tetapkan izin yang sesuai untuk Siemens Databus konfigurasi Anda.

Setelah membuat Siemens Databus konfigurasi Anda, Anda dapat menginstal aplikasi AWS IoT SiteWise Edge di aplikasi AndaSiemens Industrial Edge Management. Untuk informasi selengkapnya, lihat [Instal aplikasi ke Siemens perangkat](#).

Anda juga dapat secara opsional mengonfigurasi tujuan dan filter jalur untuk Siemens Industrial Edge gateway Anda. Untuk informasi selengkapnya, lihat [Tujuan dan filter jalur](#).

Akses aplikasi AWS IoT SiteWise Edge

Untuk mendapatkan akses ke aplikasi AWS IoT SiteWise Edge aktifSiemens Industrial Edge, [kirim email yang](#) meminta akses ke tim dukungan SiteWise Edge.

Sertakan informasi berikut di email Anda:

- Nama dan informasi kontak Anda
- Nama perusahaan
- Siemens Industrial EdgeID penyewa

Instal aplikasi ke Siemens perangkat

Setelah Anda mendapatkan akses ke aplikasi AWS IoT SiteWise Edge dengan mengirim email ke tim dukungan SiteWise EdgeSiemens Industrial Edge, tetapkan aplikasi ke instance. Siemens Industrial Edge Management Kemudian, Anda dapat menginstal aplikasi AWS IoT SiteWise Edge di perangkat Anda.

Untuk menginstal aplikasi AWS IoT SiteWise Edge

1. Verifikasi bahwa Docker intisari yang disediakan dalam Siemens Industrial Edge Management cocok dengan versi terbaru yang tercantum dalam. [AWS IoT SiteWise Changelog aplikasi tepi](#)

Untuk informasi selengkapnya tentang menemukan nilai Docker intisari untuk Siemens, lihat [Mengelola aplikasi](#) di Siemens Industrial Edge Perangkat dokumentasi. Siemens

Siemens Industrial Edge Management mendukung satu versi aplikasi AWS IoT SiteWise Edge sekaligus. Ambil langkah ini untuk memastikan bahwa Anda menggunakan versi terbaru aplikasi sebelum menginstal aplikasi AWS IoT SiteWise Edge di Siemens Industrial Edge perangkat Anda.

2. Tetapkan aplikasi AWS IoT SiteWise Edge ke Siemens Industrial Edge Management. Untuk informasi selengkapnya, lihat [Mengelola aplikasi](#) di bagian Manajemen Tepi Industri pada Siemens dokumentasi.
3. Dalam Edge Management, telusuri katalog untuk AWS IoT SiteWise Edge dan pilih.
4. Pilih Instal.

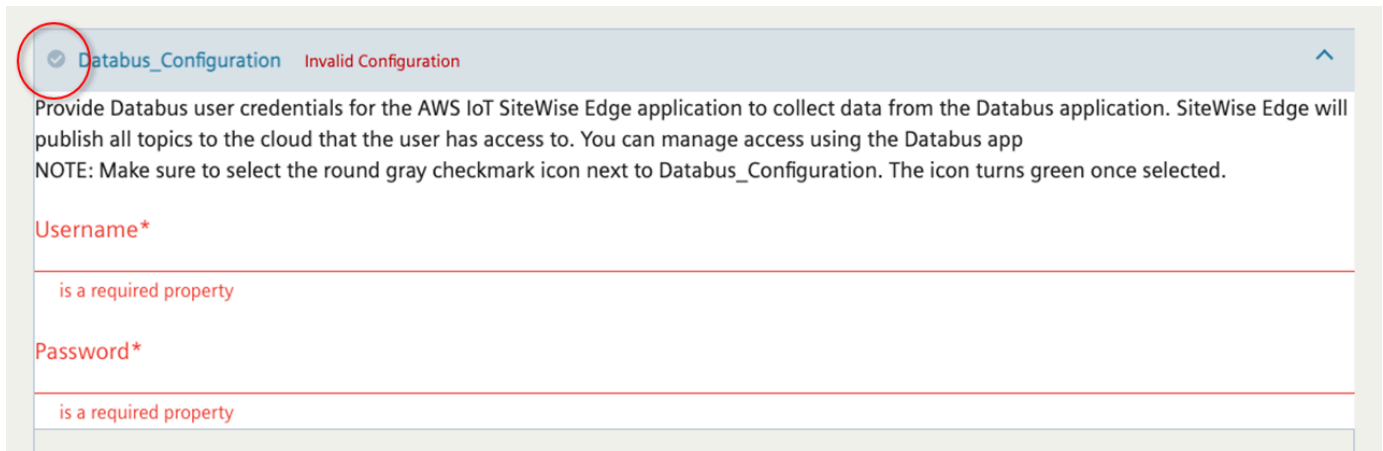
Note

Jika tombol Hubungi Kami ditampilkan, pilih tombol tersebut, dan ikuti langkah-langkah untuk meminta akses ke aplikasi AWS IoT SiteWise Edge aktif Siemens Industrial Edge. Untuk informasi selengkapnya, lihat [Akses aplikasi AWS IoT SiteWise Edge](#).

5. Pilih Databus_Configuration di opsi Konfigurasi Skema.
6. Masukkan Nama Pengguna dan Kata Sandi untuk konfigurasi Databus. Untuk informasi lebih lanjut tentang membuat Siemens Databus user, lihat [Buat Siemens Databus user untuk aplikasi](#).
7. Pilih ikon tanda centang abu-abu bulat kecil di sebelah Databus_Configuration untuk mengubah warna ikon menjadi hijau.

Note

Konfigurasi input hanya berlaku jika ikon tanda centang berubah dari abu-abu menjadi hijau. Jika tidak, konfigurasi input diabaikan.



 **Databus_Configuration** Invalid Configuration 

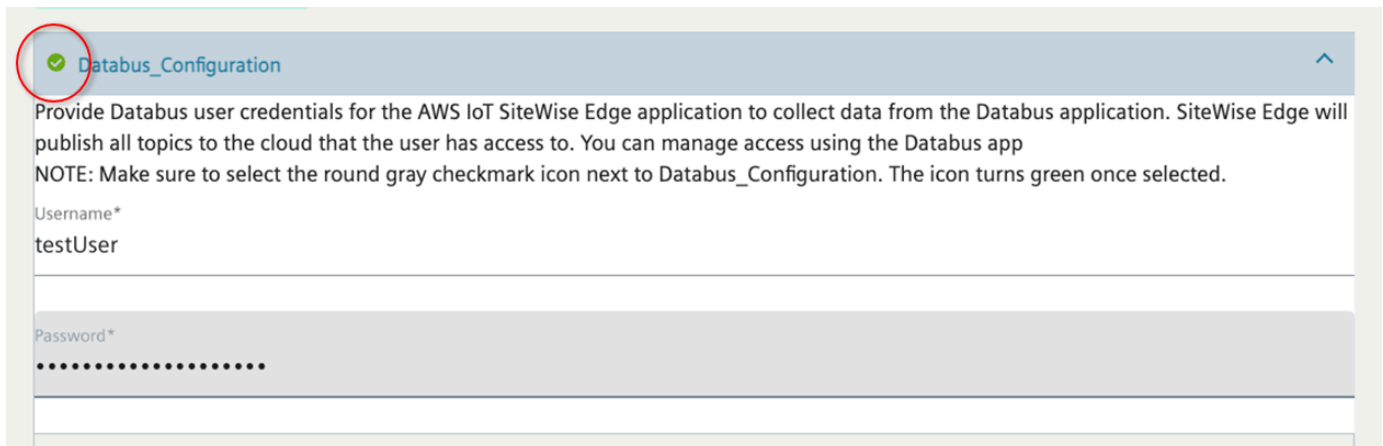
Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app
NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

is a required property

Password*

is a required property



 **Databus_Configuration** 

Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app
NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

testUser

Password*

.....

8. Pilih Berikutnya untuk pindah ke Konfigurasi Lain di mana Anda dapat mengunggah file konfigurasi gateway Anda.
9. Pilih SiteWise_Edge_Gateway_Config sebagai lokasi untuk mengunggah file konfigurasi gateway.

 Note

Pastikan Anda memilih SiteWise_Edge_Gateway_Config daripada _Edge_Support_Config_Optional. SiteWise

10. Pilih perangkat untuk menginstal aplikasi.
11. Pilih Instal sekarang.

Anda dapat mengonfigurasi komponen penerbit secara opsional untuk mengeksport data ke Cloud. AWS Untuk informasi selengkapnya, lihat [mengonfigurasi komponen AWS IoT SiteWise penerbit](#).

Untuk mengonfigurasi tujuan Siemens Industrial Edge gateway Anda, lihat [Tujuan dan filter jalur](#).

Perbarui konfigurasi aplikasi AWS IoT SiteWise Edge

Ada beberapa hal yang perlu dipertimbangkan saat memperbarui konfigurasi aplikasi AWS IoT SiteWise Edge Siemens Industrial Edge.

Note

Setiap perubahan pada konfigurasi aplikasi AWS IoT SiteWise Edge memerlukan restart aplikasi.

Alasan untuk memulai ulang aplikasi AWS IoT SiteWise Edge

- Yang baru Siemens Databus user untuk aplikasi AWS IoT SiteWise Edge.
- Perubahan pada file konfigurasi gateway (file SiteWise_Edge_Gateway_Config Anda).
- Pembaruan konfigurasi proxy (yang juga memerlukan reboot IEVD penuh)
- Untuk mengaktifkan log debug untuk masalah debugging

Mulai ulang aplikasi

1. Dalam Siemens Industrial Edge Management contoh Anda, pilih Edge Management di bagian Aplikasi Platform.
2. Pilih Aplikasi Terinstal Saya.
3. Pilih aplikasi AWS IoT SiteWise Edge.
4. Pilih Mulai Ulang.

Tujuan dan filter jalur

Tujuan di AWS IoT SiteWise Edge menyediakan cara yang fleksibel dan efisien untuk mengelola bagaimana data industri Anda mengalir dari perangkat edge ke cloud. Bagian ini menjelaskan cara mengonfigurasi tujuan, menggunakan filter jalur untuk merutekan aliran data tertentu, dan memilih jenis tujuan yang tepat untuk kasus penggunaan Anda.

Anda dapat menggunakan tujuan dan filter jalur pada gateway dan gateway V3 berkemampuan MQTT yang dihosting sendiri yang digunakan bersama dengan aplikasi Edge yang di-host. AWS

IoT SiteWise Siemens Industrial Edge Tujuan dan filter jalur tidak berfungsi dengan Aliran Klasik, gateway V2.

Topik

- [Memahami tujuan AWS IoT SiteWise Edge](#)
- [Memahami filter jalur untuk tujuan AWS IoT SiteWise Edge](#)
- [Tambahkan tujuan real-time AWS IoT SiteWise Edge](#)
- [Tambahkan tujuan AWS IoT SiteWise buffer menggunakan Amazon S3](#)
- [Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge](#)
- [Kelola tujuan AWS IoT SiteWise Edge](#)

Memahami tujuan AWS IoT SiteWise Edge

Gunakan tujuan AWS IoT SiteWise Edge untuk menentukan ke mana harus mengirim data sumber Anda. Anda dapat memilih tujuan data berdasarkan karakteristik spesifik yang Anda butuhkan, seperti efektivitas biaya, latensi rendah, atau persyaratan penyimpanan. Integrasikan data perangkat yang diambil oleh AWS IoT SiteWise, mitra kami, atau aplikasi khusus untuk mempublikasikan dan berlangganan filter jalur (topik) di edge. Anda kemudian dapat memodelkan, mentransfer, dan menyimpan data perangkat Anda di cloud.

Note

Untuk penggunaan penuh semua fitur tujuan pada gateway yang dihosting sendiri, tingkatkan ke versi terbaru SiteWise penerbit IoT dan kolektor IoT OPC UA. SiteWise Dukungan streaming dilanjutkan di aliran Klasik, gateway V2 untuk menjaga kompatibilitas dengan pengaturan yang ada. Untuk informasi selengkapnya, lihat [Aliran klasik, gateway V2 untuk Edge AWS IoT SiteWise](#)

Topik

- [Bagaimana tujuan SiteWise Edge meningkatkan manajemen data](#)
- [Jenis tujuan](#)
- [Bandingkan fungsionalitas tujuan antara versi gateway](#)
- [Batasan tujuan](#)
- [Kasus penggunaan untuk tujuan SiteWise Edge](#)

Bagaimana tujuan SiteWise Edge meningkatkan manajemen data

Ekspor data dari edge ke AWS IoT SiteWise real time, atau dalam batch menggunakan Amazon S3.

Tujuan meningkatkan fleksibilitas dan skalabilitas di AWS IoT SiteWise lingkungan Anda. Tujuan menerapkan model manajemen data terpusat, di mana sumber mempublikasikan data ke sistem pusat. Tujuan menentukan ke mana data Anda dikirim menggunakan filter jalur. Tujuan dapat berlangganan beberapa filter jalur.

Gateway berkemampuan MQTT, baik yang di-host sendiri atau berjalan Siemens Industrial Edge, menggunakan MQTT untuk komunikasi lokal dan dilengkapi dengan tujuan real-time default yang memiliki filter yang disetel ke. # Ini berarti bahwa, secara default, semua pesan tentang semua topik dipublikasikan ke tujuan AWS IoT SiteWise waktu nyata. Untuk informasi selengkapnya, lihat [Memahami filter jalur untuk tujuan AWS IoT SiteWise Edge](#). Anda dapat menambahkan satu tujuan real-time di setiap gateway.

Jenis tujuan

Saat mengonfigurasi tujuan untuk gateway, Anda memiliki dua opsi utama: penggunaan konfigurasi waktu nyata AWS IoT SiteWise, dan konfigurasi buffer menggunakan Amazon S3. Setiap jenis tujuan memiliki seperangkat pengaturan dan pertimbangannya sendiri.

AWS IoT SiteWise pengaturan waktu nyata

Pilih ini untuk mengirim data langsung ke penyimpanan AWS IoT SiteWise tingkat panas untuk memfasilitasi pengambilan dan pemantauan data secara real-time. Pengaturan real-time mengelola aliran data, terutama ketika gateway mengalami masalah konektivitas dengan cloud. Selama kehilangan koneksi, data disimpan sementara secara lokal di gateway. Setelah koneksi dibuat kembali, data yang disimpan secara otomatis dikirim ke cloud.

Anda dapat menyesuaikan berbagai aspek proses penerbitan data, seperti jumlah maksimum data yang akan disimpan secara lokal, kecepatan pengiriman data ke cloud saat penyambungan ulang, dan kapan harus menghapus data setelah penyimpanan mencapai kapasitasnya.

Untuk informasi lebih lanjut tentang tingkatan AWS IoT SiteWise penyimpanan, lihat, [Mengelola penyimpanan data di AWS IoT SiteWise](#).

AWS IoT SiteWise buffer menggunakan pengaturan Amazon S3

Jenis tujuan ini memungkinkan Anda untuk menyangga data secara lokal di gateway dan mengirimkannya secara berkala ke bucket Amazon S3 dalam batch. Data disimpan dalam format

Parket yang efisien, yang dioptimalkan untuk beban kerja analitis. Setelah data berada di Amazon S3, Anda dapat mengimpornya AWS IoT SiteWise untuk penyimpanan, pemrosesan, dan analisis.

Pilih opsi ini untuk menelan data dalam batch, dan menyimpan data historis dengan cara yang hemat biaya. Anda dapat mengonfigurasi lokasi bucket Amazon S3 pilihan Anda, dan frekuensi di mana Anda ingin data diunggah ke Amazon S3. Anda juga dapat memilih apa yang harus dilakukan dengan data setelah tertelan. AWS IoT SiteWise Anda dapat memilih untuk memiliki data yang tersedia di keduanya SiteWise dan Amazon S3 atau Anda dapat memilih untuk menghapusnya secara otomatis dari Amazon S3.

Bandingkan fungsionalitas tujuan antara versi gateway

Fitur tujuan di gateway berkemampuan MQTT merampingkan manajemen aliran data. Tujuan menyederhanakan manajemen data melalui konfigurasi terpusat perutean data ke berbagai titik akhir. Pendekatan ini menghilangkan kebutuhan untuk pengaturan aliran individu yang kompleks, membuat keseluruhan sistem lebih fleksibel dan lebih mudah dikelola.

Sebagai perbandingan, aliran Klasik, gateway V2, SiteWise Edge mentransmisikan data dari sumber data ke penerbit melalui AWS IoT Greengrass aliran, mengonfigurasi tujuan data secara individual untuk setiap sumber data.

Dengan fitur AWS IoT SiteWise tujuan, konfigurasi perutean penerbit dikonsolidasikan. Konfigurasi tujuan memungkinkan Anda mengelola tujuan dan filter jalur secara terpusat. Anda dapat dengan mudah menambahkan tujuan, mengelola filter jalur, menghapus filter atau tujuan yang tidak perlu, tergantung pada kebutuhan Anda.

Selain itu, fitur tujuan menggunakan MQTT (Message Queuing Telemetry Transport), protokol standar industri yang banyak digunakan dalam aplikasi IoT industri. Adopsi MQTT ini membantu AWS IoT SiteWise memfasilitasi integrasi yang lebih mudah dengan berbagai perangkat dan sistem.

Batasan tujuan

Batasan saat ini untuk tujuan di gateway SiteWise Edge meliputi:

- Paket pemrosesan data tidak didukung pada gateway berkemampuan MQTT.
- Dukungan tipe data terbatas pada tipe AWS IoT SiteWise data. Untuk informasi tentang mengaktifkan konversi tipe data, lihat [Mengonversi tipe data yang tidak didukung](#).

Kasus penggunaan untuk tujuan SiteWise Edge

SiteWise Tujuan tepi digunakan dalam berbagai aplikasi. Berikut adalah beberapa contoh kunci:

Otomatisasi industri, Pemantauan waktu nyata dan pemeliharaan prediktif

Dalam pengaturan industri, sensor dan perangkat di lantai pabrik dapat mempublikasikan data ke SiteWise Edge. Tujuan dapat dikonfigurasi untuk menyaring dan merutekan data yang relevan, memungkinkan pemantauan dan analisis kinerja alat berat secara real-time. Anda dapat berlangganan topik MQTT yang relevan menggunakan filter jalur, memproses data, dan kemudian mempublikasikan data yang diproses. Dengan cara ini, Anda dapat secara selektif merutekan data yang diproses ke layanan analitik AWS cloud atau sistem lokal. Produsen kemudian dapat menerapkan strategi pemeliharaan prediktif, mengoptimalkan proses produksi, dan mengurangi waktu henti.

Bangunan pintar, Efisiensi energi dan optimalisasi hunian

Sistem otomasi bangunan menghasilkan aliran data untuk memantau dan mengontrol berbagai aspek bangunan, seperti sistem HVAC, pencahayaan, dan kontrol akses. Dengan SiteWise Edge, aliran data ini dapat dicerna, diproses, dan diarahkan ke tujuan yang berbeda. Manajer fasilitas dapat mengonfigurasi tujuan untuk memfilter dan meneruskan data yang relevan, memungkinkan kemampuan canggih seperti ukuran efisiensi energi dan pengoptimalan hunian sambil memastikan privasi dan kepatuhan data.

Kasus penggunaan ini menunjukkan bagaimana fitur Destinasi di SiteWise Edge dapat dimanfaatkan di berbagai industri untuk menyerap, memproses, dan merutekan data secara efisien. Ini memungkinkan kemampuan canggih seperti pemantauan waktu nyata, pemeliharaan prediktif, efisiensi energi, dan diagnostik jarak jauh sambil memastikan privasi dan kepatuhan data.

Memahami filter jalur untuk tujuan AWS IoT SiteWise Edge

Topik

- [Persyaratan filter jalur](#)
- [Praktik terbaik untuk filter jalur](#)
- [Filter jalur untuk server OPC UA](#)
- [Karakter khusus dalam nama filter jalur](#)

Setiap tujuan dikonfigurasi untuk merutekan data ke AWS IoT SiteWise atau Amazon S3. Filter jalur memungkinkan Anda memilih data tertentu untuk difilter saat menerima pesan MQTT untuk tujuan. Filter jalur mewakili nama logis aliran data Anda, bertindak sebagai langganan ke topik MQTT yang diinginkan.

Di MQTT, data diatur ke dalam topik, yang merupakan string hierarkis yang dipisahkan oleh garis miring maju (/). / Misalnya, perangkat mungkin mempublikasikan data suhu ke topik `home/livingroom/sensor1/temperature`. Di sini, `home/livingroom/sensor1` mewakili jalur atau nama logis sensor, dan `temperature` tipe data yang diterbitkan.

Anda dapat menggunakan filter jalur untuk berlangganan topik tertentu atau berbagai topik menggunakan wildcard (+dan#). +Wildcard cocok dengan satu level dalam hierarki topik. Misalnya, `home/+sensor1/temperature` akan cocok `home/livingroom/sensor1/temperature` dan `home/bedroom/sensor1/temperature`. #Wildcard, ketika digunakan di akhir filter, cocok dengan beberapa level.

Anda juga dapat menggunakan berbagai karakter yang biasanya tidak diizinkan dalam spesifikasi MQTT dalam nama filter jalur. Karakter ini tidak berfungsi sebagai wildcard saat digunakan dalam sebuah nama. AWS IoT SiteWise mengonversi karakter ini menggunakan pengkodean untuk memastikan kepatuhan MQTT sambil mempertahankan struktur penamaan asli Anda. Fitur ini sangat berguna untuk mengakomodasi konvensi penamaan yang ada dari sistem lain. Untuk informasi selengkapnya, lihat [Karakter khusus dalam nama filter jalur](#).

Dengan hati-hati memilih filter jalur yang sesuai, Anda dapat mengontrol data mana yang dikirim ke tujuan tertentu. Sesuaikan aliran data dengan persyaratan sistem IoT Anda menggunakan filter jalur.

Persyaratan filter jalur

Saat memasukkan filter jalur menggunakan Konsol AWS IoT SiteWise, ingatlah hal berikut:

- Filter jalur dibatasi oleh baris baru, dengan setiap baris mewakili filter jalur terpisah.
- Filter jalur individu dapat memiliki antara 1 dan 65.535 byte.
- Filter jalur tidak bisa kosong.
- Nilai nol (U+0000) tidak diperbolehkan.
- Anda dapat memasukkan hingga 100 filter jalur atau 65.535 karakter sekaligus, batas mana pun yang tercapai terlebih dahulu.
- Batas keseluruhan adalah 20.000 filter jalur untuk semua tujuan di gateway digabungkan.

- Anda dapat menggunakan %, #, +, dan \$ karakter dalam nama filter jalur, namun AWS IoT SiteWise secara otomatis mengonversinya menjadi pengkodean URI.

Praktik terbaik untuk filter jalur

Saat membuat filter jalur untuk AWS IoT SiteWise tujuan Anda, pertimbangkan strategi berikut untuk mengelola data Anda secara efektif.

- Susun filter Anda untuk mencerminkan hierarki perangkat Anda. Misalnya, dalam pengaturan manufaktur `factory/+/machine/#`, menangkap data dari semua mesin di berbagai lini produksi.
- Gunakan level tertentu untuk jenis, lokasi, atau fungsi perangkat. Misalnya, `factory/assembly-line/robot/temperature`. Atau, dalam pertanian cerdas, `farm/+/crop/+/moisture`, untuk memantau tingkat kelembaban untuk berbagai tanaman di berbagai bidang.
- Leverage wildcard secara strategis: Gunakan + untuk variasi pada satu tingkat dan # untuk menangkap semua level berikutnya. Misalnya `building/+/+/energy-consumption`, melacak penggunaan energi di berbagai zona dan lantai di sebuah gedung. Ini mengasumsikan yang pertama + menangkap semua lantai dan yang kedua + menangkap semua zona.
- Menyeimbangkan kekhususan dan fleksibilitas dengan membuat filter yang cukup spesifik untuk menangkap data yang relevan tetapi cukup fleksibel untuk mengakomodasi perubahan di masa depan. Misalnya, `site/+/equipment-type/+/measurement` memungkinkan untuk penambahan situs baru atau jenis peralatan tanpa mengubah struktur filter.

Uji filter Anda secara menyeluruh untuk memastikan filter tersebut menangkap data yang diinginkan dan selaras dengan arsitektur dan sasaran sistem IoT Anda.

Filter jalur untuk server OPC UA

Untuk server OPC UA, filter jalur Anda harus sesuai dengan nama tag OPC UA. Tingkat akhir filter jalur Anda harus sama persis dengan nama tag OPC UA. Misalnya, jika tag OPC UA Anda `Device1.Temperature`, filter jalur Anda mungkin `factory/line1/Device1.Temperature`. Anda dapat menggunakan wildcard di level sebelumnya, seperti `factory/+/Device1.Temperature` untuk menangkap tag di beberapa lini produksi. Jika Anda memiliki karakter khusus dalam nama filter jalur Anda, lihat [Karakter khusus dalam nama filter jalur](#) untuk informasi selengkapnya.

Karakter khusus dalam nama filter jalur

AWS IoT SiteWise mengakomodasi karakter yang biasa digunakan dalam protokol industri seperti OPC UA, yang biasanya tidak diizinkan dalam nama topik MQTT standar. Fitur ini memfasilitasi integrasi sistem industri yang lebih lancar dengan arsitektur berbasis MQTT.

Note

Meskipun penanganan karakter khusus kami sangat membantu untuk integrasi dan migrasi, disarankan untuk menyelaraskan dengan konvensi penamaan MQTT standar untuk implementasi baru bila memungkinkan untuk memastikan kompatibilitas yang lebih luas.

Saat menerima data dari sumber industri, AWS IoT SiteWise menormalkan nama topik menggunakan pengkodean URI untuk karakter khusus:

- %menjadi %25 (dikodekan terlebih dahulu sebagai karakter pelarian)
- # menjadi %23
- + menjadi %2B
- \$menjadi %24 (hanya ketika di awal topik)

Pengkodean ini memastikan bahwa data sumber yang berisi karakter MQTT khusus ini dapat digunakan dengan aman sebagai nama topik MQTT sambil mempertahankan konvensi penamaan industri asli.

Example : Karakter khusus dalam nama filter jalur

Berikut adalah contoh bagaimana nama topik industri mungkin muncul di filter AWS IoT SiteWise jalur:

- Factory1/Line#2/Sensor+3 menjadi Factory1/Line%232/Sensor%2B3
- Plant%A/Unit\$1/Temp menjadi Plant%25A/Unit%241/Temp
- Site1/#Section/+Node menjadi Site1/%23Section/%2BNode

Saat membuat langganan atau melihat nama topik di AWS IoT SiteWise, Anda akan melihat versi asli yang tidak dikodekan. Pengkodean ditangani secara otomatis untuk memastikan kepatuhan MQTT.

Tambahkan tujuan real-time AWS IoT SiteWise Edge

Jenis tujuan real-time memungkinkan Anda untuk mengalirkan data IoT langsung dari perangkat dan gateway Anda ke AWS IoT SiteWise penyimpanan secara real-time. Opsi ini sangat ideal untuk kasus penggunaan yang memerlukan konsumsi dan pemrosesan data segera saat dihasilkan, tanpa perlu batching atau buffering. Anda hanya dapat memiliki satu tujuan real-time yang dikonfigurasi di setiap gateway, karena itu mengalirkan data secara terus AWS IoT SiteWise menerus.

Note

Duplikat TQVs dapat mengakibatkan pengisian ganda.

Untuk menambahkan tujuan real-time

Gunakan AWS IoT SiteWise konsol atau AWS CLI untuk menambahkan tujuan waktu nyata ke gateway berkemampuan SiteWise Edge MQTT Anda.

Console

1. Buka [konsol AWS IoT SiteWise](#).
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway yang ingin Anda tambahkan tujuan.
4. Di bagian Tujuan, pilih Tambah tujuan.
5. Pada halaman Tambah tujuan, masukkan Detail tujuan:
 - a. Nama untuk tujuan Anda di bidang Nama tujuan.
 - b. Pilih AWS IoT SiteWise waktu nyata untuk tipe Tujuan.
6. Konfigurasi urutan penerbitan gateway dengan menyetel urutan Penerbitan ke Publikasikan data lama terlebih dahulu atau Publikasikan data terbaru terlebih dahulu. Secara default, gateway menerbitkan data tertua terlebih dahulu.
7. Gunakan waktu tunggu batch maksimum untuk mengatur waktu maksimum penayang menunggu sebelum mengirim sekumpulan data ke AWS IoT SiteWise. Pengaturan ini berlaku untuk setiap alias. Data disimpan secara lokal hingga:
 - Waktu yang ditentukan telah berlalu, atau
 - 10 time-quality-value (TQV) entri diterima untuk alias

Kondisi mana pun yang terpenuhi terlebih dahulu memicu batch untuk dikirim ke cloud.

8. Untuk mengompres data yang diunggah, pilih kotak centang Aktifkan kompresi saat mengunggah data. Membiarkan gateway mengompres data Anda sebelum mengunggahnya ke cloud mengurangi penggunaan bandwidth.
9. Untuk memfilter data penayang yang kedaluwarsa, pilih kotak centang Kecualikan data kedaluwarsa. Pilihan ini hanya mengirimkan data aktif dan saat ini ke AWS IoT SiteWise.
10. Di bidang Cutoff period, masukkan frekuensi di mana data harus dianggap kedaluwarsa dalam kumpulan data Anda. Anda dapat menentukan apakah data dihitung dalam hitungan menit atau hari. Periode batas minimum adalah lima menit. Periode batas maksimum adalah tujuh hari.
11. Konfigurasi pengaturan penyimpanan lokal secara opsional:
 - a. Mengatur frekuensi periode Retensi — Jumlah waktu gateway secara lokal menyimpan data yang lebih tua dari periode cutoff. Periode retensi minimum adalah satu menit.

Periode retensi maksimum adalah 30 hari dan lebih besar dari atau sama dengan periode rotasi.
 - b. Atur periode Rotasi - Interval waktu untuk menentukan saat menyimpan data yang lebih lama dari periode cutoff untuk satu file. Gateway mentransfer satu batch data ke direktori lokal berikut pada akhir setiap periode rotasi: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`.

Retensi harus lebih besar dari satu menit dan sama dengan periode retensi.
 - c. Berikan nilai kapasitas Penyimpanan (GB) untuk mengatur ukuran maksimum data yang disimpan secara lokal dalam GB. Jika data melebihi ukuran penyimpanan lokal maksimum yang ditentukan, gateway mulai menghapus data tertua terlebih dahulu. Gateway terus menghapus hingga ukuran data yang disimpan secara lokal sama dengan atau kurang dari kuota.

Kapasitas penyimpanan harus lebih besar dari atau sama dengan satu GB.

12. Tambahkan filter jalur ke tujuan Anda. Untuk informasi selengkapnya, lihat [Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge](#).

Untuk informasi selengkapnya, lihat [Jenis tujuan](#).

AWS CLI

Example : Buat tujuan AWS IoT SiteWise real-time baru

Gunakan [UpdateGatewayCapabilityConfiguration](#) API untuk mengonfigurasi penerbit.

Atur parameter capabilityNamespace ke `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "maxBatchWaitTime": "10s"
      },
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "#"
            ]
          }
        }
      ]
    }
  ]
}
```

Untuk memperbarui tujuan AWS IoT SiteWise real-time yang ada, pertama-tama gunakan `DescribeGatewayCapabilityConfiguration` API untuk menemukan `destinationId`.

Example : Perbarui tujuan AWS IoT SiteWise waktu nyata

Gunakan [UpdateGatewayCapabilityConfiguration](#) API untuk mengonfigurasi penerbit.

Atur parameter `capabilityNamespace` ke `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "dropPolicy": {
          "cutoffAge": "7d",
          "exportPolicy": {
            "retentionPeriod": "7d",
            "rotationPeriod": "6h",
            "exportSizeLimitGB": 10
          }
        },
        "maxBatchWaitTime": "10s"
      },
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "#"
            ]
          }
        }
      ]
    }
  ]
}
```

Opsi konfigurasi berikut khusus untuk gateway menggunakan namespace.

`iotsitewise:publisher:3`

sources

Mendefinisikan sumber data untuk mentransfer data dari peralatan industri Anda ke AWS IoT SiteWise. Untuk gateway berkemampuan MQTT, gunakan MQTT.

Tipe: Array objek

Wajib: Ya

destinations

Mendefinisikan ke mana harus mengirim data. Tujuan dapat real-time atau buffer menggunakan Amazon S3. Setidaknya satu objek tujuan diperlukan, tetapi Anda dapat menambahkan array kosong. Anda dapat memiliki satu tujuan real-time untuk setiap gateway. Untuk informasi selengkapnya, lihat [Memahami tujuan AWS IoT SiteWise Edge](#).

Tipe: Array objek

Wajib: Ya

id

Pengenal unik untuk tujuan. Anda dapat memberikan ID tujuan yang ada atau membiarkannya kosong. Jika Anda tidak menentukan ID maka UUID dihasilkan secara default.

Tipe: String

Wajib: Tidak

type

Jenis tujuan. Pilihan meliputi: SITEWISE_REALTIME dan SITEWISE_BUFFERED.

- SITEWISE_REALTIME— Kirim data langsung ke AWS IoT SiteWise penyimpanan secara real-time.
- SITEWISE_BUFFERED— Kirim data ke Amazon S3 dalam batch dalam format Parquet, lalu impor ke penyimpanan. AWS IoT SiteWise

Tipe: String

Diperlukan: Ya

name

Nama unik untuk tujuan.

Tipe: String

Diperlukan: Ya

`config`

Konfigurasi khusus untuk jenis tujuan dalam format JSON. Konfigurasi bervariasi antara tujuan real-time dan buffer.

Tipe: Objek

Wajib: Ya

`PublishingOrder`

Menentukan urutan di mana data dipublikasikan. Data diterbitkan berdasarkan stempel waktunya. Pilihan termasuk `TIME_ORDER` dan `RECENT_DATA`.

- `TIME_ORDER`(default) - Menerbitkan data lama terlebih dahulu.
- `RECENT_DATA`— Menerbitkan data terbaru terlebih dahulu.

Tipe: String

Wajib: Tidak

`Aktifkan Kompresi`

Saat diatur ke `true`, aktifkan kompresi data sebelum dikirim ke AWS IoT SiteWise. Membiarkan gateway mengompres data Anda sebelum mengunggahnya ke cloud mengurangi penggunaan bandwidth. Nilai default-nya adalah `true`.

Tipe: Boolean

Wajib: Tidak

`DropPolicy`

Mendefinisikan cara menangani data yang lebih lama.

Tipe: Objek

Wajib: Tidak

- `cutoffAge`

Usia maksimum data yang akan dipublikasikan ditentukan dalam hari, jam, dan menit. Misalnya, 7d atau 1d7h16m. Data yang lebih tua dari yang Anda tentukan tidak dikirim ke AWS IoT SiteWise.

Data yang lebih awal dari periode cutoff tidak dipublikasikan ke cloud. Usia batas harus antara lima menit dan tujuh hari.

Anda dapat menggunakan `m`, `h`, dan `d` ketika Anda menentukan batas usia. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari.

Tipe: String

Diperlukan: Ya

- `exportPolicy`

Mendefinisikan cara menangani data yang melebihi batas usia.

Tipe: Objek

Wajib: Tidak

- `retentionPeriod`

Gateway SiteWise Edge Anda menghapus data apa pun di tepi yang lebih awal dari periode cutoff dari penyimpanan lokal setelah disimpan untuk periode retensi yang ditentukan. Periode retensi harus antara satu menit dan 30 hari, dan lebih besar dari atau sama dengan periode rotasi.

Anda dapat menggunakan `m`, `h`, dan `d` ketika Anda menentukan periode retensi. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari.

Tipe: String

Wajib: Tidak

- `rotationPeriod`

Interval waktu untuk mengumpulkan dan menyimpan data yang lebih awal dari periode cutoff ke satu file. Gateway SiteWise Edge mentransfer satu batch data ke direktori lokal berikut pada akhir setiap periode rotasi: `/greengrass/v2/work/`

`aws.iot.SiteWiseEdgePublisher/exports`. Periode rotasi harus lebih besar dari satu menit, dan sama dengan atau kurang dari periode retensi.

Anda dapat menggunakan `m`, `h`, dan `d` ketika Anda menentukan periode rotasi. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari.

Tipe: String

Wajib: Tidak

- `exportSizeLimitGB`

Ukuran maksimum yang diizinkan dari data yang disimpan secara lokal, dalam GB. Jika kuota ini dilanggar, gateway SiteWise Edge mulai menghapus data paling awal hingga ukuran data yang disimpan secara lokal sama dengan atau kurang dari kuota. Nilai parameter ini harus lebih besar dari atau sama dengan 1.

Tipe: Integer

Wajib: Tidak

`maxBatchWaitTime`

Menetapkan waktu maksimum bagi penayang untuk menunggu sebelum mengirim sekumpulan data ke AWS IoT SiteWise. Pengaturan ini berlaku untuk setiap alias. Data disimpan secara lokal hingga:

- Waktu yang ditentukan telah berlalu, atau
- 10 time-quality-value (TQV) entri diterima untuk alias

Gunakan `m`, `h`, dan `d` untuk menentukan batas waktu. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari.

Tipe: String

Wajib: Tidak

`filters`

Filter untuk diterapkan ke data. Setidaknya diperlukan satu filter.

Tipe: String

Diperlukan: Ya

type

Jenis filter. Gunakan PATH.

Tipe: String

Diperlukan: Ya

config

Konfigurasi khusus untuk jenis filter dalam format JSON. Setidaknya satu objek diperlukan, tetapi array bisa kosong.

Tipe: Objek

Wajib: Ya

- paths

Array filter jalur. Untuk informasi selengkapnya, lihat [Memahami filter jalur untuk tujuan AWS IoT SiteWise Edge](#). Jalur defaultnya adalah#.

Tipe: Array string

Wajib: Ya

Tambahkan tujuan AWS IoT SiteWise buffer menggunakan Amazon S3

Jenis tujuan buffer memungkinkan Anda menghemat biaya konsumsi AWS IoT SiteWise jika Anda tidak memerlukan data secara real-time. Ini memungkinkan Anda untuk menyimpan sementara data IoT Anda di bucket Amazon S3 sebelum mengimpornya ke AWS IoT SiteWise. Atau, Anda cukup mengunggah data Anda ke S3 untuk penyimpanan, terlepas dari apakah Anda berencana untuk mengimpornya. AWS IoT SiteWise ini berguna untuk mengelompokkan dan menyangga data dari perangkat dan gateway Anda sebelum memasukkannya ke dalam AWS IoT SiteWise. Dengan opsi ini, data diunggah ke bucket S3 yang ditentukan dalam format Parquet pada frekuensi yang dikonfigurasi. Anda kemudian dapat mengimpor data ini ke AWS IoT SiteWise penyimpanan untuk analisis dan pemrosesan lebih lanjut.

Untuk menambahkan tujuan yang di-buffer menggunakan Amazon S3

Gunakan AWS IoT SiteWise konsol atau AWS CLI untuk menambahkan tujuan yang menyangga data menggunakan Amazon S3 ke gateway Edge SiteWise Anda.

Console

Gunakan AWS Management Console untuk menambahkan AWS IoT SiteWise tujuan buffer menggunakan Amazon S3.

1. Buka [konsol AWS IoT SiteWise](#).
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway yang ingin Anda tambahkan tujuan.
4. Di bagian Tujuan, pilih Tambah tujuan.
5. Pada halaman Tambah tujuan, masukkan Detail tujuan:
 - a. Nama untuk tujuan Anda di bidang Nama tujuan.
 - b. Pilih AWS IoT SiteWise buffer menggunakan Amazon S3 untuk tipe Tujuan. AWS IoT SiteWise buffer menggunakan Amazon S3 mengirimkan data ke Amazon Simple Storage Service dalam batch, dalam format Parquet, dan kemudian mengimpor data ke penyimpanan. AWS IoT SiteWise
6. Masukkan URL Amazon S3 untuk lokasi tempat Anda ingin menyimpan data gateway Anda. Anda dapat menelusuri jalur dengan memilih Browse S3. Setelah bucket ditambahkan, Anda juga dapat melihat bucket dengan memilih View.
7. Tentukan seberapa sering gateway Anda harus mengunggah data ke Amazon S3 dengan memasukkan kerangka waktu dan memilih kenaikan waktu untuk frekuensi unggah Data. Nilai frekuensi harus lebih besar dari 0 dan kurang dari atau sama dengan 30 hari.
8. Dalam pengaturan penyimpanan data, tentukan apa yang harus dilakukan dengan data gateway Anda setelah mengimpornya. AWS IoT SiteWise Ada dua keputusan yang harus diambil terkait penyimpanan data:
 - Jika Anda ingin menyalin data yang diimpor ke AWS IoT SiteWise penyimpanan, pilih kotak centang Salin data ke penyimpanan. Opsi ini menduplikasi data yang diimpor dari bucket AWS IoT SiteWise Amazon S3 yang dikonfigurasi ke dalam penyimpanan.
 - Jika Anda memilih untuk mengimpor data dari bucket Amazon S3 ke AWS IoT SiteWise penyimpanan, Anda juga dapat menentukan apakah data yang diimpor harus dihapus setelah impor selesai. Pilih kotak centang Hapus data dari Amazon S3 untuk menghapus tanggal impor dari bucket Amazon S3 yang dikonfigurasi setelah mengimpornya ke penyimpanan. AWS IoT SiteWise
9. Tambahkan filter jalur ke tujuan Anda. Untuk informasi selengkapnya, lihat [Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge](#).

AWS CLI

Example : Buat AWS IoT SiteWise tujuan baru yang di-buffer menggunakan Amazon S3

Gunakan [UpdateGatewayCapabilityConfiguration](#) API untuk mengonfigurasi penerbit.

Atur parameter capabilityNamespace ke `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_BUFFERED",
      "name": "your-s3-destination-name",
      "config": {
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
        "publishPolicy": {
          "publishFrequency": "15m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,
          "enableDeleteAfterImport": true,
          "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
        }
      },
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  ]
}
```

Example : Perbarui AWS IoT SiteWise tujuan yang di-buffer menggunakan Amazon S3

Untuk memperbarui tujuan AWS IoT SiteWise real-time yang ada, pertama-tama gunakan DescribeGatewayCapabilityConfiguration API untuk menemukan destinationId.

Namespace penerbit: iotsitewise:publisher:3

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_BUFFERED",
      "name": "your-s3-destination-name",
      "config": {
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
        "publishPolicy": {
          "publishFrequency": "15m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,
          "enableDeleteAfterImport": true,
          "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
        }
      },
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  ]
}
```

```
]
}
```

Opsi konfigurasi berikut khusus untuk gateway berkemampuan MQTT menggunakan namespace. `iotsitewise:publisher:3`

`sources`

Mendefinisikan sumber data untuk mentransfer data dari peralatan industri Anda ke AWS IoT SiteWise. Untuk gateway berkemampuan MQTT, gunakan. MQTT

Tipe: Array objek

Wajib: Ya

`destinations`

Mendefinisikan ke mana harus mengirim data. Tujuan dapat real-time atau buffer menggunakan Amazon S3. Setidaknya satu objek tujuan diperlukan, tetapi Anda dapat menambahkan array kosong. Anda dapat memiliki satu tujuan real-time untuk setiap gateway. Untuk informasi selengkapnya, lihat [Memahami tujuan AWS IoT SiteWise Edge](#).

Tipe: Array objek

Wajib: Ya

`id`

Pengenal unik untuk tujuan. Anda dapat memberikan ID tujuan yang ada atau membiarkannya kosong agar ID baru dibuat secara otomatis untuk tujuan tersebut.

Tipe: String

Wajib: Tidak

`type`

Jenis tujuan. Pilihan meliputi: `SITewise_REALTIME` dan `SITewise_BUFFERED`. Pilih `SITewise_BUFFERED`.

- `SITewise_REALTIME`(default) - Kirim data langsung ke AWS IoT SiteWise penyimpanan secara real-time. Untuk informasi selengkapnya, lihat [Tambahkan tujuan real-time AWS IoT SiteWise Edge](#).

- SITEWISE_BUFFERED— Kirim data ke Amazon S3 dalam batch dalam format Parquet, lalu impor ke penyimpanan. AWS IoT SiteWise

Tipe: String

Diperlukan: Ya

name

Nama unik untuk tujuan.

Tipe: String

Diperlukan: Ya

config

Konfigurasi khusus untuk jenis tujuan dalam format JSON. Konfigurasi bervariasi antara tujuan real-time dan buffer.

Tipe: Objek

Wajib: Ya

targetBucketArn

Ember ARN untuk dipublikasikan ke. Pilih yang sama Wilayah AWS untuk keduanya AWS IoT SiteWise dan Amazon S3. Jika awalan dipilih, itu harus memiliki antara 1-255 karakter.

 Note

AWS IoT SiteWise, termasuk gateway, akan memiliki akses ke seluruh bucket S3 yang ditentukan. Sebaiknya gunakan bucket khusus untuk konsumsi data buffer.

Tipe: String

Diperlukan: Ya

publishPolicy

Rincian kebijakan penerbitan.

Tipe: Objek

Wajib: Ya

`publishFrequency`

Frekuensi penerbitan gateway SiteWise Edge ke bucket Amazon S3. Frekuensi unggah data ke Amazon S3 harus lebih besar dari 0 menit dan kurang dari atau sama dengan 30 hari. Anda dapat menggunakan `m`, `h`, dan `d` ketika Anda menentukan usia frekuensi penerbitan. Catatan yang `m` mewakili menit, `h` mewakili jam, dan `d` mewakili hari. Nilai default adalah 15 menit.

Tipe: String

Diperlukan: Ya

`localSizeLimitGB`

Ukuran maksimum file yang ditulis ke disk lokal dalam GB. Jika ambang batas ini dilanggar, penerbit menerbitkan semua data buffer ke tujuannya.

Jenis: Integer

Wajib: Ya

`siteWiseImportPolicy`

Rincian kebijakan impor untuk mengimpor data ke AWS IoT SiteWise.

Tipe: Objek

Wajib: Ya

`enableSiteWiseStorageImport`

Setel ini `true` untuk mengimpor data dari bucket Amazon S3 ke AWS IoT SiteWise penyimpanan. Awalnya membuat salinan data di AWS IoT SiteWise. Kemudian, jika Anda mengatur `enableDeleteAfterImport` ke `true`, data di S3 dihapus setelah menyalin ke AWS IoT SiteWise. Implikasi harga berlaku. Nilai default-nya adalah `true`.

Tipe: Boolean

Wajib: Ya

`enableDeleteAfterImport`

Setel ini `true` untuk menghapus file di bucket Amazon S3 setelah tertelan ke penyimpanan. AWS IoT SiteWise Nilai default-nya adalah `true`.

Tipe: Boolean

Wajib: Ya

`bulkImportJobRoleArn`

ARN dari peran IAM yang AWS IoT SiteWise mengasumsikan untuk membaca data buffer dari Amazon S3 selama konsumsi data. Peran ini digunakan saat perangkat edge memanggil AWS IoT SiteWise APIs untuk memulai proses impor massal.

Note

Jika `enableSiteWiseStorageImport` diatur ke `true`, parameter ini diperlukan.

Tipe: String

Wajib: Tidak

Tambahkan filter jalur untuk tujuan Anda. Untuk informasi selengkapnya, lihat [Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge](#).

Tambahkan filter jalur ke tujuan AWS IoT SiteWise Edge

Tambahkan filter jalur ke tujuan. Filter jalur menggunakan sintaks topik MQTT, di mana `#` adalah karakter wildcard yang cocok dengan sejumlah level, dan `+` merupakan karakter wildcard yang cocok dengan satu level. Anda dapat menambahkan beberapa tujuan ke gateway, masing-masing dengan set filter jalurnya sendiri yang berlangganan telemetry peralatan Anda.

Siemens Industrial Edgegateway menggunakan awalan untuk kompatibilitas. Untuk informasi selengkapnya, lihat [Awalan untuk filter jalur](#).

Console

Untuk menambahkan filter jalur

1. Buka [konsol AWS IoT SiteWise](#).
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway yang ingin Anda tambahkan filter jalur.
4. Di bagian Filter jalur di bawah Tambahkan tujuan, pilih Tambahkan filter jalur.
5. Masukkan filter jalur yang Anda inginkan untuk berlangganan tujuan ini. Anda dapat menggunakan karakter wildcard (#dan+) untuk berlangganan beberapa jalur.
6. Pilih Tambahkan filter jalur untuk menambahkan filter jalur ke daftar.
7. Ulangi langkah untuk menambahkan filter jalur tambahan, jika diperlukan.
8. Setelah Anda menambahkan semua filter jalur yang diperlukan, pilih Buat.

AWS CLI for self-hosted gateways

Example : Konfigurasi filter jalur

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "home/+/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "building/#"
        ]
      }
    }
  ]
}
```

AWS CLI for Siemens IEGateways

Example : Konfigurasi awalan untuk filter jalur

Tangkap semua data dengan menggunakan awalan data (ie/d) dan metadata (ie/m) untuk setiap filter jalur.

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "ie/d/home/+/sensor12/temperature",
          "ie/m/home/livingroom/sensor12/temperature",
          "ie/d/home/livingroom/sensor13/temperature2",
          "ie/m/home/livingroom/sensor13/temperature2",
          "ie/d/building/#",
          "ie/m/building/#"
        ]
      }
    }
  ]
}
```

Note

Salin filter jalur antar tujuan dengan mengunduh daftar filter jalur. Untuk informasi selengkapnya, lihat [Unduh semua filter jalur di tujuan \(konsol\)](#).

Unggah filter jalur secara massal

Untuk mengunggah filter jalur secara massal, gunakan CSV atau file teks. AWS IoT SiteWise secara otomatis menghapus duplikat yang tepat saat Anda mengunggah file. Misalnya, windfarm/site1/

dan windfarm/site1/ merupakan duplikat persis yang AWS IoT SiteWise menangkap karena string persis sama. Duplikat sebagian tidak dihapus dan mengakibatkan biaya tambahan. Misalnya, windfarm/# dan windfarm/site1 merupakan topik yang tumpang tindih karena windfarm/site1 sudah dicakup oleh. windfarm/#

Note

Hindari duplikat untuk mencegah biaya tambahan. File yang diunggah harus dalam format.csv atau .txt. Itu tidak dapat berisi header apa pun dan harus terdiri dari satu kolom. Di kolom, daftarkan filter jalur Anda, dengan setiap filter pada baris terpisah. Tidak ada informasi lain yang harus disertakan dalam file.

Persyaratan unggah file

Ini adalah persyaratan filter jalur tambahan.

- Anda dapat mengunggah satu file.csv atau .txt. Format file lainnya tidak didukung.
- File CSV (.csv) tidak dapat memiliki header dan hanya boleh berisi satu kolom.
- Anda dapat memiliki satu filter jalur pada setiap baris.
- File yang diunggah tidak dapat kosong.
- Saat menggunakan # sebagai wildcard, itu harus menjadi karakter terakhir dalam filter topik. Misalnya, topic/# atau sebagai karakter mandiri pada tingkat topik tertentu. Namun, perhatikan bahwa juga # dapat digunakan sebagai karakter biasa dalam nama tingkat topik, seperti factory/machine#1/topic. Untuk informasi lebih lanjut lihat [Karakter khusus dalam nama filter jalur](#)
- Anda juga dapat menggunakan + karakter. Misalnya, gunakan factory/+/temp untuk mendapatkan semua suhu untuk pabrik, bukan factory/machine2/temp dan factory/machine3/temp secara individual.

Kelola tujuan AWS IoT SiteWise Edge

Setelah menambahkan tujuan, Anda dapat melakukan berbagai operasi untuk mengelolanya, seperti mengedit konfigurasi tujuan, menghapus tujuan, dan mengelola filter jalur.

Mengedit tujuan

Pilih tombol radio di sebelah tujuan dalam tabel dan pilih tombol Edit untuk mengedit tujuan.

Console

Untuk mengedit tujuan menggunakan AWS IoT SiteWise konsol

1. Buka [konsol AWS IoT SiteWise](#).
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway yang sesuai.
4. Di bagian Tujuan, pilih tujuan yang ingin Anda edit, lalu pilih Edit.
5. Ubah tujuan dan kemudian pilih Simpan.

AWS CLI

Untuk mengedit tujuan menggunakan AWS CLI

- Anda dapat mengedit tujuan dengan memodifikasi informasi konfigurasi kemampuan JSON.

```
aws iotsitewise update-gateway-capability-configuration \
--gateway-id your-gateway-id \
--capability-namespace "iotsitewise:publisher:3" \
--capability-configuration '{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-updated-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "dropPolicy": {
          "cutoffAge": "10d",
          "exportPolicy": {
            "retentionPeriod": "10d",
            "rotationPeriod": "6h",
            "exportSizeLimitGB": 10
          }
        }
      }
    }
  ],
}
```

```
        "maxBatchWaitTime": "15s"
      },
      "filters": [
        {
          ...
        }
      ]
    }
  ]
}'
```

Note

Anda tidak dapat memperbarui tujuan type `ataucapability-namespace`. Misalnya, Anda tidak dapat beralih dari jenis `SITewise_REALTIME` ke `SITewise_BUFFERED`. Anda dapat memiliki satu tujuan real-time untuk setiap gateway berkemampuan MQTT.

Menghapus tujuan

Jika Anda tidak lagi membutuhkan tujuan, Anda dapat menghapusnya dari gateway SiteWise Edge Anda.

Console

Untuk menghapus tujuan menggunakan AWS IoT SiteWise konsol

1. Buka [konsol AWS IoT SiteWise](#).
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway yang sesuai.
4. Di bagian Tujuan, pilih tujuan yang ingin Anda hapus lalu pilih Hapus. Layar konfirmasi muncul.
5. Untuk mengonfirmasi pilihan Anda untuk menghapus tujuan, ketik "hapus" di kotak konfirmasi.

AWS CLI

Untuk menghapus tujuan menggunakan AWS CLI

- Hapus konfigurasi kapabilitas gateway dengan menentukan ID gateway dan memodifikasi konfigurasi kemampuan untuk menghapus tujuan yang ingin Anda hapus.

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --capability-configuration '{  
    "sources": [  
      {  
        "type": "MQTT"  
      }  
    ],  
    "destinations": []  
  }'
```

Note

Array tujuan dapat kosong ([]), tetapi objek tujuan itu sendiri harus disertakan dalam konfigurasi kemampuan.

Unduh semua filter jalur di tujuan (konsol)

Unduh file CSV yang berisi semua filter jalur Anda di AWS IoT SiteWise konsol. Anda dapat menggunakan daftar filter jalur yang diunduh untuk dengan mudah berbagi daftar filter jalur antar tujuan gateway.

Untuk mengunduh file CSV dari semua filter jalur menggunakan konsol AWS IoT SiteWise

1. Buka [konsol AWS IoT SiteWise](#).
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway yang berisi filter jalur Anda.
4. Pilih Tambahkan tujuan atau Edit tujuan.
5. Arahkan ke bagian Filter jalur dan pilih Unduh CSV.

Note

File CSV mencakup semua filter jalur di tujuan tertentu, terlepas dari mana yang Anda pilih dari daftar filter jalur.

Mengedit filter jalur

Anda dapat mengedit filter jalur individual untuk menyempurnakan data mana yang diterima tujuan Anda.

Console

Menggunakan AWS IoT SiteWise konsol, Anda dapat mengedit setiap filter jalur individu dalam setiap kotak teks masing-masing.

Untuk mengedit filter jalur menggunakan AWS IoT SiteWise konsol

1. Buka [konsol AWS IoT SiteWise](#).
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway yang berisi filter jalur Anda.
4. Pilih tujuan yang sesuai.
5. Pilih Edit.
6. Pilih kotak teks untuk baris yang berisi filter jalur yang ingin Anda edit.
7. Perbarui teks filter jalur, memastikan bahwa kotak centang filter jalur yang diedit dipilih.
8. Pilih Simpan.

AWS CLI

Untuk mengedit filter jalur untuk tujuan menggunakan AWS CLI, pertama-tama ambil konfigurasi saat ini, modifikasi, lalu perbarui menggunakan `update-gateway-capability-configuration` perintah.

Untuk mengedit filter jalur menggunakan AWS CLI

1. Ambil konfigurasi kemampuan saat ini:

```
aws iotsitewise describe-gateway-capability-configuration \
```

```
--gateway-id your-gateway-id \  
--capability-namespace "iotsitewise:publisher:3" \  
--query "capabilityConfiguration"
```

2. Edit JSON untuk memodifikasi filter jalur sesuai kebutuhan.
3. Perbarui konfigurasi kemampuan dengan filter jalur yang dimodifikasi:

```
aws iotsitewise update-gateway-capability-configuration \  
--gateway-id your-gateway-id \  
--capability-namespace "iotsitewise:publisher:3" \  
--capability-configuration json-containing-your-updated-path-filters
```

Hapus filter jalur

Anda dapat menghapus filter jalur untuk tujuan untuk mengontrol data yang diterimanya dari sumber MQTT dan pipeline pemrosesan data.

Console

Untuk menghapus filter jalur menggunakan AWS IoT SiteWise konsol

1. Buka [konsol AWS IoT SiteWise](#).
2. Di navigasi kiri, pilih gateway Edge di bagian Edge.
3. Pilih gateway yang berisi filter jalur Anda.
4. Pilih tujuan yang sesuai.
5. Pilih Edit.
6. Pada layar Edit tujuan, di bagian Filter jalur, pilih satu atau beberapa filter jalur yang akan dihapus.
7. Pilih Hapus. Pesan konfirmasi penghapusan muncul. Jika ingin melanjutkan dengan menghapus filter jalur, pilih Hapus di layar konfirmasi.

AWS CLI

Untuk menghapus tujuan menggunakan AWS CLI

- Hapus filter jalur dengan menghapusnya dari konfigurasi kemampuan.

```
aws iotsitewise update-gateway-capability-configuration \  
--gateway-id your-gateway-id \  
--capability-namespace "iotsitewise:publisher:3" \  
--capability-configuration json-containing-your-updated-path-filters
```

```

--gateway-id your-gateway-id \
--capability-namespace "iotsitewise:publisher:3" \
--capability-configuration '{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        ...
      },
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "/path1",
              "/path2",
              "/delete-a-path-to-remove-it"
            ]
          }
        }
      ]
    }
  ]
}

```

 Note

Array filter dapat kosong ([]), tetapi objek filter itu sendiri harus disertakan dalam konfigurasi kemampuan.

Kelola SiteWise gateway Edge

Anda dapat menggunakan operasi AWS IoT SiteWise konsol dan API untuk mengelola gateway AWS IoT SiteWise Edge. Anda juga dapat menggunakan aplikasi [AWS OpsHub for AWS IoT SiteWise for Windows](#) untuk mengelola beberapa aspek gateway SiteWise Edge Anda dari perangkat lokal Anda.

Kami sangat menyarankan Anda menggunakan AWS IoT SiteWise aplikasi AWS OpsHub for untuk memantau penggunaan disk pada perangkat lokal Anda. Anda juga dapat memantau CloudWatch metrik `Gateway.AvailableDiskSpace` dan `Gateway.UsedPercentageDiskSpace` Amazon dan membuat alarm untuk mendapatkan pemberitahuan saat ruang disk semakin rendah. Untuk informasi selengkapnya tentang CloudWatch alarm Amazon, lihat [Membuat CloudWatch alarm berdasarkan ambang batas statis](#).

Pastikan perangkat Anda memiliki cukup ruang untuk data yang akan datang. Ketika Anda akan kehabisan ruang di perangkat lokal Anda, layanan secara otomatis menghapus sejumlah kecil data dengan stempel waktu tertua untuk memberi ruang bagi data yang akan datang.

Untuk memeriksa apakah layanan menghapus data Anda, lakukan hal berikut:

1. Masuk ke AWS IoT SiteWise aplikasi AWS OpsHub untuk.
2. Pilih Pengaturan.
3. Untuk Log, tentukan rentang waktu, lalu pilih Unduh.
4. Buka zip file log.
5. Jika file log berisi pesan berikut, layanan menghapus data Anda: *number*byte data telah dihapus untuk mencegah penyimpanan gateway SiteWise Edge kehabisan ruang.

Kelola gateway SiteWise Edge Anda dengan AWS IoT SiteWise konsol

Anda dapat menggunakan AWS IoT SiteWise konsol untuk mengonfigurasi, memperbarui, dan memantau semua gateway SiteWise Edge di akun Anda AWS .

[Anda dapat melihat gateway SiteWise Edge Anda dengan menavigasi ke halaman Edge Gateways di konsol.AWS IoT SiteWise](#) Untuk mengakses halaman detail gateway Edge untuk gateway tertentu, pilih nama gateway Edge.

Dari tab Ikhtisar halaman detail gateway Edge, Anda dapat melakukan hal berikut:

- Di bagian Sumber data, perbarui konfigurasi sumber data dan konfigurasi sumber data tambahan
- Pilih Buka CloudWatch metrik untuk melihat jumlah titik data yang dicerna per sumber data di konsol metrik CloudWatch
- Di bagian kemampuan Edge, tambahkan paket data ke gateway SiteWise Edge Anda dengan mengklik Edit
- Di bagian konfigurasi Gateway, lihat status konektivitas gateway SiteWise Edge Anda
- Di bagian konfigurasi Publisher, lihat status sinkronisasi gateway SiteWise Edge dan konfigurasi komponen AWS IoT SiteWise penerbit

Dari tab Pembaruan pada halaman detail gateway Edge, Anda dapat melihat komponen saat ini dan versi paket yang digunakan ke gateway Edge. Ini juga tempat Anda menerapkan versi baru, saat tersedia.

Kelola gateway SiteWise Edge menggunakan for AWS OpsHubAWS IoT SiteWise

Anda menggunakan AWS IoT SiteWise aplikasi AWS OpsHub for untuk mengelola dan memantau gateway SiteWise Edge yang dihosting sendiri. Aplikasi ini menyediakan opsi pemantauan dan manajemen berikut:

- Di bawah Ikhtisar, Anda dapat melakukan hal berikut:
 - Lihat detail gateway SiteWise Edge yang membantu Anda mendapatkan wawasan tentang data perangkat gateway SiteWise Edge Anda, mengidentifikasi masalah, dan meningkatkan kinerja gateway SiteWise Edge.
 - Lihat portal SiteWise Monitor yang memantau data dari server lokal dan peralatan di tepi. Untuk informasi selengkapnya, lihat [Apa yang ada AWS IoT SiteWise Monitor](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- Di bawah Health, ada dasbor yang menampilkan data dari gateway SiteWise Edge Anda. Pakar domain, seperti insinyur proses, dapat menggunakan dasbor untuk melihat ikhtisar perilaku gateway SiteWise Edge.
- Di bawah Aset, lihat aset yang digunakan ke perangkat lokal dan nilai terakhir yang dikumpulkan atau dihitung untuk properti aset.
- Di bawah Pengaturan, Anda dapat melakukan hal berikut:

- Jika Paket Pemrosesan Data diinstal, lihat informasi konfigurasi gateway SiteWise Edge dan sinkronkan sumber daya dengan AWS Cloud.
- Unduh file otentikasi yang dapat Anda gunakan untuk mengakses gateway SiteWise Edge dengan menggunakan alat lain.
- Unduh log yang dapat Anda gunakan untuk memecahkan masalah gateway SiteWise Edge.
- Lihat AWS IoT SiteWise komponen yang digunakan ke gateway SiteWise Edge.

Important

Berikut ini diperlukan untuk digunakan AWS OpsHub untuk AWS IoT SiteWise:

- Perangkat lokal Anda dan AWS IoT SiteWise aplikasi AWS OpsHub for harus terhubung ke jaringan yang sama.
- Paket pemrosesan data harus diaktifkan.

Untuk mengelola gateway SiteWise Edge menggunakan AWS OpsHub

1. Unduh dan instal aplikasi [AWS OpsHubAWS IoT SiteWise untuk Windows](#).
2. Buka aplikasi .
3. Jika Anda tidak memiliki kredensi lokal yang disiapkan untuk gateway Anda, ikuti langkah-langkah di bawah ini [Akses gateway SiteWise Edge Anda menggunakan kredensyal sistem operasi lokal](#) untuk mengaturnya.
4. Anda dapat masuk ke gateway SiteWise Edge Anda dengan kredensi Linux atau Lightweight Directory Access Protocol (LDAP) Anda. Untuk masuk ke gateway SiteWise Edge Anda, lakukan salah satu hal berikut:

Linux

1. Untuk nama host atau alamat IP, masukkan nama host atau alamat IP perangkat lokal Anda.
2. Untuk Otentikasi, pilih Linux.
3. Untuk nama pengguna, masukkan nama pengguna sistem operasi Linux Anda.
4. Untuk Kata Sandi, masukkan kata sandi sistem operasi Linux Anda.
5. Pilih Masuk.

LDAP

1. Untuk nama host atau alamat IP, masukkan nama host atau alamat IP perangkat lokal Anda.
2. Untuk Otentikasi, pilih LDAP.
3. Untuk nama Pengguna, masukkan nama pengguna LDAP Anda.
4. Untuk Kata Sandi, masukkan kata sandi LDAP Anda.
5. Pilih Masuk.

Akses gateway SiteWise Edge Anda menggunakan kredensyal sistem operasi lokal

Selain Lightweight Directory Access Protocol (LDAP), Anda dapat menggunakan kredensi Linux atau Windows untuk mengakses gateway Edge yang dihosting sendiri SiteWise .

Important

Untuk mengakses gateway SiteWise Edge Anda dengan kredensi Linux, Anda harus mengaktifkan paket pemrosesan data untuk gateway SiteWise Edge Anda.

Akses gateway SiteWise Edge Anda menggunakan kredensyal sistem operasi Linux

Langkah-langkah berikut mengasumsikan bahwa Anda menggunakan perangkat dengan Ubuntu. Jika Anda menggunakan distribusi Linux yang berbeda, lihat dokumentasi yang relevan untuk perangkat Anda.

Untuk membuat kumpulan pengguna Linux

1. Untuk membuat grup admin, jalankan perintah berikut.

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

Pengguna dalam SWE_ADMIN_GROUP grup dapat mengizinkan akses admin untuk gateway SiteWise Edge.

2. Untuk membuat grup pengguna, jalankan perintah berikut.


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

Pengguna dalam SWE_USER_GROUP grup dapat mengizinkan akses hanya-baca untuk gateway SiteWise Edge.

3. Untuk menambahkan pengguna ke grup admin, jalankan perintah berikut. Ganti *user-name* dan *password* dengan nama pengguna dan kata sandi yang ingin Anda tambahkan.

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

4. Untuk menambahkan pengguna ke salah satu SWE_ADMIN_GROUP atau SWE_USER_GROUP, ganti *user-name* dengan nama pengguna yang Anda tambahkan pada langkah sebelumnya.

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

Anda sekarang dapat menggunakan nama pengguna dan kata sandi untuk masuk ke gateway SiteWise Edge pada AWS IoT SiteWise aplikasi AWS OpsHub for.

Akses gateway SiteWise Edge Anda menggunakan kredensial Windows

Langkah-langkah berikut mengasumsikan bahwa Anda menggunakan perangkat dengan Windows.

Important

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Buat kebijakan kata sandi yang kuat dengan setidaknya 12 karakter dan kombinasi huruf besar, huruf kecil, angka, dan simbol. Selain itu, tetapkan aturan Windows Firewall untuk memungkinkan lalu lintas masuk pada port 443 dan untuk memblokir lalu lintas masuk di semua port lainnya.

Untuk membuat kumpulan pengguna Windows Server

1. Jalankan PowerShell sebagai administrator.
 - a. Di server Windows tempat Anda ingin menginstal SiteWise Edge Gateway, masuk sebagai administrator.
 - b. Masukkan PowerShell di bilah pencarian Windows.

- c. Di hasil pencarian, klik kanan pada PowerShell aplikasi Windows. Pilih Jalankan sebagai Administrator.
2. Untuk membuat grup admin, jalankan perintah berikut.

```
net localgroup SWE_ADMIN_GROUP /add
```

Anda harus menjadi pengguna dalam SWE_ADMIN_GROUP grup untuk mengizinkan akses admin untuk gateway SiteWise Edge.

3. Untuk membuat grup pengguna, jalankan perintah berikut.

```
net localgroup SWE_USER_GROUP /add
```

Anda harus menjadi pengguna dalam SWE_USER_GROUP grup untuk mengizinkan akses siapa-saja untuk gateway SiteWise Edge.

4. Untuk menambahkan pengguna, jalankan perintah berikut. Ganti *user-name* dan *password* dengan nama pengguna dan kata sandi yang ingin Anda buat.

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

5. Untuk menambahkan pengguna ke grup admin, jalankan perintah berikut. Ganti *user-name* dengan nama pengguna yang ingin Anda tambahkan.

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

Anda sekarang dapat menggunakan nama pengguna dan kata sandi untuk masuk ke gateway SiteWise Edge pada AWS IoT SiteWise aplikasi AWS OpsHub for.

Kelola sertifikat gateway SiteWise Edge

Anda dapat menggunakan SiteWise Monitor dan aplikasi pihak ketiga, seperti Grafana, di perangkat gateway SiteWise Edge Anda. Aplikasi ini memerlukan koneksi TLS ke layanan. SiteWise Edge gateway saat ini menggunakan sertifikat yang ditandatangani sendiri. Jika Anda menggunakan browser untuk membuka aplikasi, seperti portal SiteWise Monitor, Anda mungkin menerima peringatan untuk sertifikat yang tidak tepercaya.

Berikut ini menunjukkan cara mengunduh sertifikat tepercaya dari AWS IoT SiteWise aplikasi AWS OpsHub for.

1. Masuk ke aplikasi.
2. Pilih Pengaturan.
3. Untuk Otentikasi, pilih Unduh sertifikat.

Berikut ini mengasumsikan bahwa Anda menggunakan Google Chrome atau FireFox. Jika Anda menggunakan browser yang berbeda, lihat dokumentasi yang relevan untuk browser Anda. Untuk menambahkan sertifikat yang Anda unduh pada langkah sebelumnya ke browser, lakukan salah satu hal berikut:

- Jika Anda menggunakan Google Chrome, ikuti [Menyiapkan sertifikat](#) di dokumentasi Bantuan Google Chrome Enterprise.
- Jika Anda menggunakan Firefox, ikuti To [Load the Certificate ke Mozilla atau Firefox Browser](#) dalam dokumentasi Oracle.

Ubah versi paket komponen gateway SiteWise Edge

Anda dapat menggunakan AWS IoT SiteWise konsol untuk mengubah versi paket komponen di gateway SiteWise Edge Anda.

Untuk mengubah versi paket komponen gateway SiteWise Edge

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Gateway.
3. Pilih gateway SiteWise Edge yang ingin Anda ubah versi paketnya.
4. Di bawah konfigurasi Gateway, pilih Lihat versi perangkat lunak.
5. Pada halaman Edit versi perangkat lunak, untuk paket yang ingin Anda perbarui versi, pilih versi yang ingin Anda gunakan dan pilih Deploy.
6. Pilih Selesai.

Daftar SiteWise gateway Edge

Console

Untuk membuat daftar SiteWise gateway Edge

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di panel navigasi, pilih gateway Edge.
3. Lihat daftar semua gateway SiteWise Edge Anda.

AWS CLI

Untuk membuat daftar gateway Anda menggunakan AWS CLI, ikuti langkah-langkah ini:

- Gunakan perintah `list-gateways` untuk melihat semua gateway Anda:

```
aws iotsitewise list-gateways
```

Perintah ini mengembalikan daftar gateway Anda dengan nama IDs, nama, dan informasi lainnya.

Anda juga dapat menentukan parameter pagination:

```
aws iotsitewise list-gateways --max-results 10 --next-token your-token
```

Untuk informasi selengkapnya, lihat [daftar-gateway](#) di Referensi Perintah. AWS CLI

Jelaskan gateway SiteWise Edge

Console

Untuk melihat detail gateway

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih nama gateway yang ingin Anda lihat detailnya.
4. Lihat detail gateway di halaman detail gateway Edge.

AWS CLI

Untuk mendapatkan informasi terperinci tentang penggunaan gateway tertentu AWS CLI, ikuti langkah-langkah berikut:

- Gunakan perintah `deskripsi-gateway` dengan ID gateway:

```
aws iotsitewise describe-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Perintah ini mengembalikan informasi rinci tentang gateway.

Untuk informasi selengkapnya, lihat [deskripsi-gateway](#) di Referensi Perintah. AWS CLI

Buat gateway SiteWise Edge

Console

Untuk membuat gateway SiteWise Edge

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih Buat gateway.
4. Masukkan nama untuk gateway Anda.
5. Pilih grup Greengrass untuk gateway Anda.
6. Secara opsional, tambahkan tag ke gateway Anda.
7. Pilih Buat.

AWS CLI

Untuk membuat SiteWise gateway IoT baru menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Gunakan perintah `create-gateway` untuk membuat gateway baru:

```
aws iotsitewise create-gateway \
  --gateway-name "NewSiteWiseGateway" \
  --gateway-platform '{
    "greengrass": {
      "groupArn": "arn:aws:greengrass:us-east-1:123456789012:group/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
    }
  }' \
```

```
--tags '{
  "Environment": "Production",
  "Location": "Factory1"
}'
```

Perintah ini mengembalikan ID gateway baru dan ARN:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

Untuk informasi selengkapnya, lihat [create-gateway](#) di Command Reference. AWS CLI

Perbarui gateway SiteWise Edge

Console

Untuk memperbarui gateway SiteWise Edge

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway yang ingin Anda perbarui.
4. Pilih Edit.
5. Perbarui nama gateway atau pengaturan lain sesuai kebutuhan.
6. Pilih Simpan.

AWS CLI

Untuk memperbarui gateway yang ada menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Gunakan perintah `update-gateway` untuk memperbarui nama gateway:

```
aws iotsitewise update-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --gateway-name "UpdatedGatewayName"
```

Perintah ini tidak menghasilkan output saat berhasil.

Untuk informasi selengkapnya, lihat [update-gateway](#) di Command Reference. AWS CLI

Perbarui konfigurasi kemampuan gateway

Console

Untuk memperbarui konfigurasi kemampuan gateway

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih nama gateway yang ingin Anda perbarui.
4. Di bagian Sumber data, pilih Edit.
5. Perbarui konfigurasi sumber data sesuai kebutuhan.
6. Pilih Simpan.

AWS CLI

Untuk memperbarui konfigurasi kemampuan gateway menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Gunakan `update-gateway-capability-configuration` perintah untuk memperbarui konfigurasi kemampuan:

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1" \
  --capability-configuration '{
    "sources": [
      {
        "name": "OPC-UA Server",
        "endpoint": {
          "certificateTrust": {
            "type": "TrustAny"
          },
          "endpointUri": "opc.tcp://10.0.0.1:4840",
```

```

        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
            "type": "Anonymous"
        }
    },
    "measurementDataStreamPrefix": ""
}
]
}'

```

Perintah ini mengembalikan namespace kemampuan dan status sinkronisasi:

```

{
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilitySyncStatus": "CONFIGURING"
}

```

Untuk informasi selengkapnya, lihat [update-gateway-capability-configuration](#) di Referensi AWS CLI Perintah.

Tag sumber daya gateway

Console

Untuk menandai sumber daya gateway

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih nama gateway yang ingin Anda tag.
4. Pilih tab Tanda.
5. Pilih Kelola tanda.
6. Pilih Tambahkan tag baru dan masukkan kunci dan nilai untuk setiap tag.
7. Pilih Simpan.

AWS CLI

Untuk menambahkan tag ke gateway menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Gunakan perintah `tag-resource` untuk menambahkan tag ke gateway:

```
aws iotsitewise tag-resource \  
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \  
  --tags '{  
    "Department": "Operations",  
    "Project": "FactoryAutomation"  
  }'
```

Perintah ini tidak menghasilkan output saat berhasil.

Untuk informasi selengkapnya, lihat [tag-resource](#) di AWS CLI Command Reference.

Tag daftar untuk gateway

Console

Untuk membuat daftar tag untuk gateway

- Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
- Di panel navigasi, pilih gateway Edge.
- Pilih nama gateway yang ingin Anda lihat tag.
- Pilih tab Tanda.
- Lihat daftar tag yang terkait dengan gateway.

AWS CLI

Untuk mencantumkan tag yang terkait dengan gateway menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Gunakan `list-tags-for-resource` perintah untuk membuat daftar tag untuk gateway:

```
aws iotsitewise list-tags-for-resource \  

```

```
--resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

Perintah ini mengembalikan tag yang terkait dengan gateway:

```
{
  "tags": {
    "Environment": "Production",
    "Location": "Factory1",
    "Department": "Operations",
    "Project": "FactoryAutomation"
  }
}
```

Untuk informasi selengkapnya, lihat [list-tags-for-resource](#) di Referensi AWS CLI Perintah.

Hapus tag dari gateway

Console

Untuk menghapus tag dari gateway

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih nama gateway yang ingin Anda hapus tag.
4. Pilih tab Tanda.
5. Pilih Kelola tanda.
6. Pilih ikon hapus (X) di samping setiap tag yang ingin Anda hapus.
7. Pilih Simpan.

AWS CLI

Untuk menghapus tag dari gateway menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Gunakan perintah `untag-resource` untuk menghapus tag dari gateway:

```
aws iotsitewise untag-resource \
```

```
--resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \
--tag-keys '["Project", "Department"]'
```

Perintah ini tidak menghasilkan output saat berhasil.

Untuk informasi selengkapnya, lihat [untag-resource](#) di Command Reference. AWS CLI

Perbarui versi AWS IoT SiteWise komponen

Perbarui komponen AWS IoT SiteWise gateway pada perangkat AWS IoT Greengrass inti Anda untuk memastikan akses Anda ke fitur terbaru, peningkatan kinerja, dan tambalan keamanan.

Untuk memperbarui AWS IoT SiteWise komponen pada AWS IoT Greengrass

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih gateway Edge.
3. Pilih gateway yang akan diedit dan pilih Edit.
4. Di Edge Capabilities, di bawah versi Perangkat Lunak, pilih Pembaruan perangkat lunak tersedia. Halaman Edit versi perangkat lunak muncul.
5. Pilih versi komponen.

Note

Disarankan untuk memilih versi terbaru yang tersedia. Menjaga komponen gateway up-to-date membantu Anda mempertahankan fungsionalitas optimal untuk pengumpulan dan pemrosesan data industri.

6. Pilih Deploy. Ini memulai AWS IoT Greengrass V2 penerapan untuk memperbarui AWS IoT SiteWise komponen di gateway.

Hapus gateway SiteWise Edge

Console

Untuk menghapus gateway SiteWise Edge

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway yang ingin Anda hapus.
4. Pilih Hapus.
5. Untuk mengonfirmasi bahwa Anda ingin menghapus gateway, ketik “hapus” lalu pilih Hapus di jendela yang muncul.

AWS CLI

Untuk menghapus gateway menggunakan AWS CLI, ikuti langkah-langkah berikut:

1. Buat daftar gateway Anda untuk mengidentifikasi ID gateway gateway yang ingin Anda hapus.

```
aws iotsitewise list-gateways
```

Perintah ini mengembalikan daftar gateway Anda dengan nama IDs, nama, dan informasi lainnya:

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

```
]
}
```

2. Hapus gateway dengan menentukan ID-nya:

```
aws iotsitewise delete-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Perintah ini tidak menghasilkan output saat berhasil.

 Note

Saat Anda menghapus gateway, beberapa file gateway tetap berada di sistem file gateway Anda.

3. Untuk memverifikasi bahwa gateway telah dihapus, Anda dapat mencantumkan gateway Anda lagi:

```
aws iotsitewise list-gateways
```

Gateway yang dihapus seharusnya tidak lagi muncul dalam daftar.

Untuk informasi selengkapnya, lihat [delete-gateway di Referensi](#) AWS CLI Perintah.

Cadangkan dan pulihkan gateway SiteWise Edge

Topik ini mencakup cara memulihkan gateway SiteWise Edge dan mencadangkan data metrik Anda. Jika Anda mengalami masalah dengan gateway SiteWise Edge yang rusak pada mesin yang sama dan perlu memecahkan masalah, baca AWS IoT SiteWise dokumentasi [SiteWise Pemecahan Masalah gateway Edge](#).

 Note

Panduan yang dibahas dalam topik ini adalah untuk gateway SiteWise Edge yang diinstal pada AWS IoT Greengrass V2 versi 2.1.0 atau lebih tinggi.

Backup harian data metrik

Membuat cadangan adalah penting, jika Anda ingin mentransfer atau mengembalikan data pada mesin baru. Mencadangkan data Anda sangat mengurangi risiko hilangnya data operasi selama proses transfer atau pemulihan.

Bagian ini berlaku untuk gateway yang menggunakan paket pemrosesan data. Untuk informasi selengkapnya tentang paket pemrosesan data, lihat [Konfigurasi model aset untuk pemrosesan data di SiteWise Edge](#).

Jalur folder influxdb adalah sebagai berikut:

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

Kami menyarankan Anda membuat cadangan seluruh folder dengan semua yang ada di bawahnya.

Kami menyarankan Anda secara berkala mencadangkan data metrik Anda dari 1.0 SiteWise Edge ke hard drive eksternal atau ke AWS cloud.

Kembalikan gateway SiteWise Edge

Sebelum mencoba memulihkan gateway SiteWise Edge, pastikan bahwa semua perangkat tepi yang terhubung ke gateway dihentikan atau terputus.

Gunakan prosedur berikut untuk memulihkan gateway SiteWise Edge:

1. Gunakan skrip instalasi yang diunduh saat Anda membuat gateway SiteWise Edge untuk memulihkan gateway SiteWise Edge pada mesin baru. Baca [Menginstal perangkat lunak gateway SiteWise Edge pada prosedur perangkat lokal Anda](#) untuk mengatur gateway SiteWise Edge.

Jika Anda kehilangan atau tidak dapat menemukan skrip instalasi, silakan hubungi [AWS Customer Support](#).

2. Setelah gateway SiteWise Edge diinstal, masuk ke [AWS IoT Greengrass konsol](#).

3. Untuk menerapkan ulang komponen, navigasikan ke Kelola lalu di bawah AWS IoT Greengrass perangkat pilih Perangkat inti.
4. Di tabel perangkat AWS IoT Greengrass inti pilih perangkat inti yang sesuai dengan gateway SiteWise Edge Anda.
5. Setelah berada di halaman perangkat, buka tab Deployment dan pilih ID Deployment Anda, ini akan membuka halaman Deployment dengan ID yang Anda pilih.
6. Setelah Anda berada di halaman Deployments, di kanan atas tekan tombol Tindakan, dan pilih opsi Revisi. untuk memulai penerapan baru. Konfigurasikan penerapan. Jika Anda ingin mempertahankan penerapan apa adanya, lewati ke Review and Deploy.
7. Tunggu Status Deployment menjadi Completed.

 Note

Ini juga akan memakan waktu beberapa menit untuk semua komponen di SiteWise Edge untuk sepenuhnya setup dan berjalan.

Kembalikan AWS IoT SiteWise data

Gunakan prosedur berikut untuk memulihkan data pada mesin baru.

1. Salin influxdb folder ke mesin baru.
2. Hentikan SiteWise EdgeProcessor komponen, dengan menjalankan perintah berikut di terminal Anda:

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. Temukan jalur tempat Anda mencadangkan data, dan jalankan perintah berikut:

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. Mulai ulang SiteWiseEdgeProcessor komponen:

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
```

Validasi backup dan restorasi yang berhasil

Gunakan prosedur ini untuk memvalidasi data cadangan dan SiteWise restorasi gateway Edge Anda.

Note

Prosedur ini mengharuskan Anda menginstal AWS OpsHub untuk AWS IoT SiteWise. Untuk informasi selengkapnya, lihat, [Mengelola gateway SiteWise Edge menggunakan AWS OpsHub](#) for. AWS IoT SiteWise

1. Terbuka AWS OpsHub untuk AWS IoT SiteWise.

2. Pada halaman Pengaturan SiteWise Edge Gateway, periksa status setiap komponen yang tercantum dalam tabel Komponen. Verifikasi bahwa warna status berwarna hijau dan pembacaan menampilkan RUNNING.
3. Validasi data masa lalu Anda di dasbor portal untuk memeriksa apakah data sebelumnya dan data baru keduanya diatur dengan benar. Akan ada downtime antara data masa lalu dan baru. Anda harus kecuali untuk melihat durasi di mana tidak ada titik data yang dikumpulkan.

[Jika Anda mengalami masalah dengan mencadangkan atau memulihkan gateway SiteWise Edge, lihat topik pemecahan masalah berikut Memecahkan masalah gateway Edge. AWS IoT SiteWise](#)

Gerbang warisan ()AWS IoT Greengrass Version 1

Note

SiteWise Gateway tepi yang berjalan hanya AWS IoT Greengrass V1 tersedia jika Anda mulai menggunakan fitur ini sebelum 29 Juli 2021. Untuk informasi selengkapnya tentang menjalankan AWS IoT SiteWise gateway menggunakan AWS IoT Greengrass V2, lihat [Self-host gateway AWS IoT SiteWise Edge dengan AWS IoT Greengrass V2](#).

SiteWise Edge gateway sekarang secara eksklusif berjalan AWS IoT Greengrass V2, menyediakan fungsionalitas yang ditingkatkan dan peningkatan kinerja untuk aplikasi IoT industri Anda. Versi terbaru ini AWS IoT Greengrass V2 mewakili evolusi arsitektur, dibangun di atas kerangka berbasis komponen modern yang memungkinkan penyebaran perangkat lunak modular. Ini merampingkan instalasi melalui installer terpadu sambil menawarkan pengembang fleksibilitas yang lebih besar dalam menerapkan komponen khusus dan melakukan pengujian lokal. Model berbasis komponen memungkinkan manajemen sumber daya yang lebih efisien dan memperkenalkan pendekatan konfigurasi yang disederhanakan melalui resep komponen. Desain ini memfasilitasi penanganan ketergantungan yang lebih baik antar komponen, mendukung praktik penerapan berkelanjutan, dan menyediakan kemampuan CLI yang ditingkatkan untuk pengembangan lokal. Selain itu, AWS IoT Greengrass V2 memusatkan manajemen konfigurasi melalui AWS IoT Core dan memberikan fitur pencatatan dan pemantauan yang lebih baik, semuanya dilindungi oleh model izin keamanan yang lebih terperinci.

Untuk informasi lebih lanjut tentang memulai menggunakan AWS IoT Greengrass V2 gateway SiteWise Edge, lihat [AWS IoT SiteWise Persyaratan gateway yang dihosting sendiri di Edge](#) Sumber

daya ini memberikan step-by-step petunjuk tentang pengaturan gateway Anda, mengonfigurasi sumber data, dan mengelola infrastruktur IoT industri Anda.

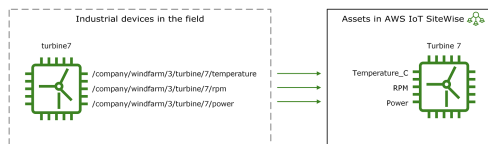
 Note

Karena AWS terus berinovasi dan meningkatkan layanan IoT-nya, disarankan untuk tetap diperbarui dengan fitur dan peningkatan terbaru. Periksa AWS IoT SiteWise dan AWS IoT Greengrass dokumentasi secara teratur untuk kemampuan baru yang dapat lebih mengoptimalkan solusi IoT industri Anda.

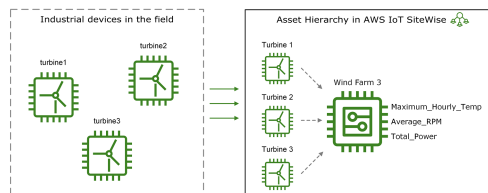
Model aset industri

Anda dapat membuat representasi virtual dari operasi industri Anda dengan AWS IoT SiteWise aset. Aset mewakili perangkat, peralatan, atau proses yang mengunggah satu atau beberapa aliran data ke Cloud. AWS Misalnya, perangkat aset dapat berupa turbin angin yang mengirimkan suhu udara, kecepatan rotasi baling-baling, dan pengukuran deret waktu keluaran daya ke properti aset di. AWS IoT SiteWise

Setiap aliran data sesuai dengan alias properti unik. Misalnya, alias `/company/windfarm/3/turbine/7/temperature` secara unik mengidentifikasi aliran data suhu yang berasal dari turbin #7 di ladang angin #3. Anda dapat mengonfigurasi AWS IoT SiteWise aset untuk mengubah data pengukuran yang masuk menggunakan ekspresi matematika, seperti mengonversi data suhu dari Celcius ke Fahrenheit.



Aset juga dapat mewakili pengelompokan perangkat yang logis, seperti seluruh ladang angin. Anda dapat mengaitkan aset dengan aset lain untuk membuat hierarki aset yang mewakili operasi industri yang kompleks. Aset dapat mengakses data dalam aset anak terkait. Dengan demikian, Anda dapat menggunakan AWS IoT SiteWise ekspresi untuk menghitung metrik agregat, seperti output daya bersih dari ladang angin.



Anda harus membuat setiap aset dari model aset. Model aset adalah struktur deklaratif yang menstandarisasi format aset Anda. Model aset menerapkan informasi yang konsisten di beberapa aset dengan jenis yang sama sehingga Anda dapat memproses data dalam aset yang mewakili grup perangkat. Pada diagram sebelumnya, Anda menggunakan model aset yang sama untuk ketiga turbin karena semua turbin memiliki satu set properti yang sama.

Anda juga dapat membuat model komponen. Model komponen adalah jenis model aset khusus yang dapat Anda sertakan dalam model aset atau model komponen lainnya. Anda dapat menggunakan

model komponen untuk menentukan sub-rakitan umum yang dapat digunakan kembali, seperti sensor, motor, dan sebagainya, yang Anda bagikan di beberapa model aset.

Setelah Anda menentukan model aset Anda, Anda dapat membuat aset industri Anda. Untuk membuat aset, pilih model ACTIVE aset untuk membuat aset dari model tersebut. Kemudian, Anda dapat mengisi informasi khusus aset seperti alias dan atribut aliran data. Pada diagram sebelumnya, Anda membuat tiga aset turbin dari satu model aset dan kemudian mengaitkan alias aliran data seperti `/company/windfarm/3/turbine/7/temperature` untuk setiap turbin.

Anda juga dapat memperbarui dan menghapus aset, model aset, dan model komponen yang ada. Saat Anda memperbarui model aset, setiap aset berdasarkan model aset tersebut mencerminkan perubahan apa pun yang Anda buat pada model yang mendasarinya. Saat Anda memperbarui model komponen, ini berlaku untuk setiap aset berdasarkan setiap model aset yang mereferensikan model komponen.

Model aset Anda mungkin sangat kompleks, misalnya saat memodelkan peralatan rumit yang memiliki banyak subkomponen. Untuk membantu menjaga agar model aset tersebut tetap teratur dan dapat dipelihara, Anda dapat menggunakan model komposit khusus untuk mengelompokkan properti terkait atau menggunakan kembali komponen bersama. Untuk informasi selengkapnya, lihat [Model komposit khusus \(komponen\)](#).

Topik

- [Status aset dan model](#)
- [Versi model aset](#)
- [Model komposit khusus \(komponen\)](#)
- [Mengatur AWS IoT SiteWise objek IDs](#)
- [Buat model aset dan komponen untuk AWS IoT SiteWise](#)
- [Buat aset untuk model aset di AWS IoT SiteWise](#)
- [Cari aset di Konsol AWS IoT SiteWise](#)
- [Perbarui nilai atribut](#)
- [Megasosiasikan dan memisahkan aset](#)
- [Perbarui aset dan model](#)
- [Hapus aset dan model di AWS IoT SiteWise](#)
- [Operasi massal dengan aset dan model](#)

Status aset dan model

Saat Anda membuat, memperbarui, atau menghapus aset, model aset, atau model komponen, perubahan membutuhkan waktu untuk disebar. AWS IoT SiteWise menyelesaikan operasi ini secara asinkron dan memperbarui status setiap sumber daya. Setiap aset, model aset, dan model komponen memiliki bidang status yang berisi status sumber daya dan pesan kesalahan apa pun, jika berlaku. Negara dapat menjadi salah satu dari nilai berikut:

- **ACTIVE**— Sumber daya aktif. Ini adalah satu-satunya status di mana Anda dapat melakukan kueri dan berinteraksi dengan aset, model aset, dan model komponen.
- **CREATING**— Sumber daya sedang dibuat.
- **UPDATING**— Sumber daya sedang diperbarui.
- **DELETING**— Sumber daya sedang dihapus.
- **PROPAGATING**— (Model aset dan model komponen saja) Perubahan menyebar ke semua sumber daya dependen (dari model aset ke aset, atau dari model komponen ke model aset).
- **FAILED**— Sumber daya gagal memvalidasi selama operasi membuat atau memperbarui, mungkin karena referensi melingkar dalam ekspresi. Anda dapat menghapus sumber daya yang ada di **FAILED** negara bagian.

Beberapa operasi membuat, memperbarui, dan menghapus AWS IoT SiteWise menempatkan aset, model aset, atau model komponen dalam keadaan selain **ACTIVE** saat operasi diselesaikan. Untuk melakukan kueri atau berinteraksi dengan sumber daya setelah Anda melakukan salah satu operasi ini, Anda harus menunggu hingga status berubah **ACTIVE**. Jika tidak, permintaan Anda gagal.

Topik

- [Periksa status aset](#)
- [Memeriksa status aset atau model komponen](#)

Periksa status aset

Anda dapat menggunakan AWS IoT SiteWise konsol atau API untuk memeriksa status aset.

Topik

- [Periksa status aset \(konsol\)](#)
- [Memeriksa status aset \(AWS CLI\)](#)

Periksa status aset (konsol)

Gunakan prosedur berikut untuk memeriksa status aset di AWS IoT SiteWise konsol.

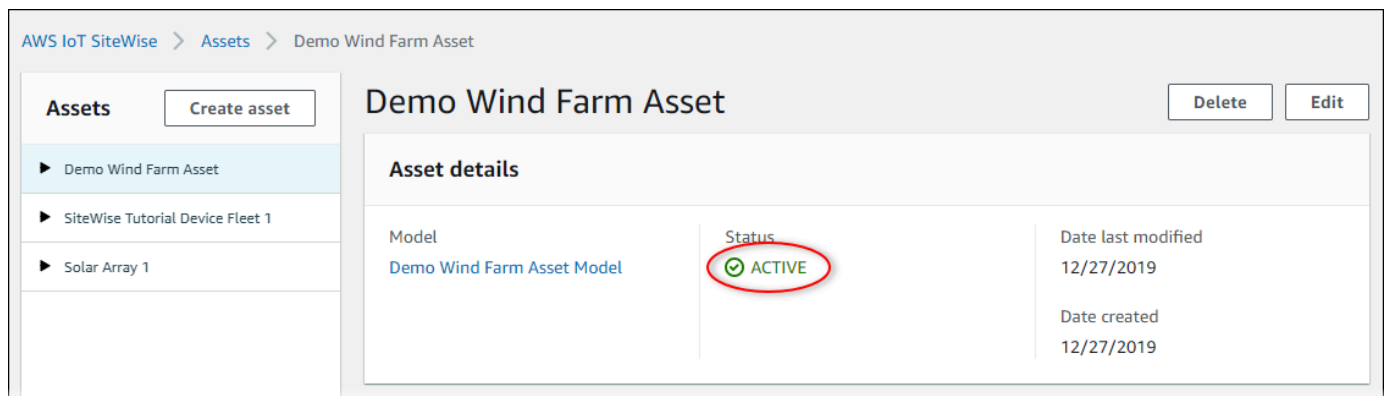
Untuk memeriksa status aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang akan diperiksa.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Temukan Status di panel detail Aset.



Memeriksa status aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk memeriksa status aset.

Untuk memeriksa status aset, gunakan [DescribeAsset](#) operasi dengan `assetId` parameter.

Untuk memeriksa status aset (AWS CLI)

- Jalankan perintah berikut untuk menggambarkan aset. Ganti *asset-id* dengan ID aset atau ID eksternal. ID eksternal adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

```
aws iotsitewise describe-asset --asset-id asset-id
```

Operasi mengembalikan respons yang berisi detail aset. Respons berisi `assetStatus` objek yang memiliki struktur berikut:

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

Status aset ada `assetStatus.state` di objek JSON.

Memeriksa status aset atau model komponen

Anda dapat menggunakan AWS IoT SiteWise konsol atau API untuk memeriksa status model aset atau model komponen.

Topik

- [Periksa status model aset atau model komponen \(konsol\)](#)
- [Periksa status model aset atau model komponen \(AWS CLI\)](#)

Periksa status model aset atau model komponen (konsol)

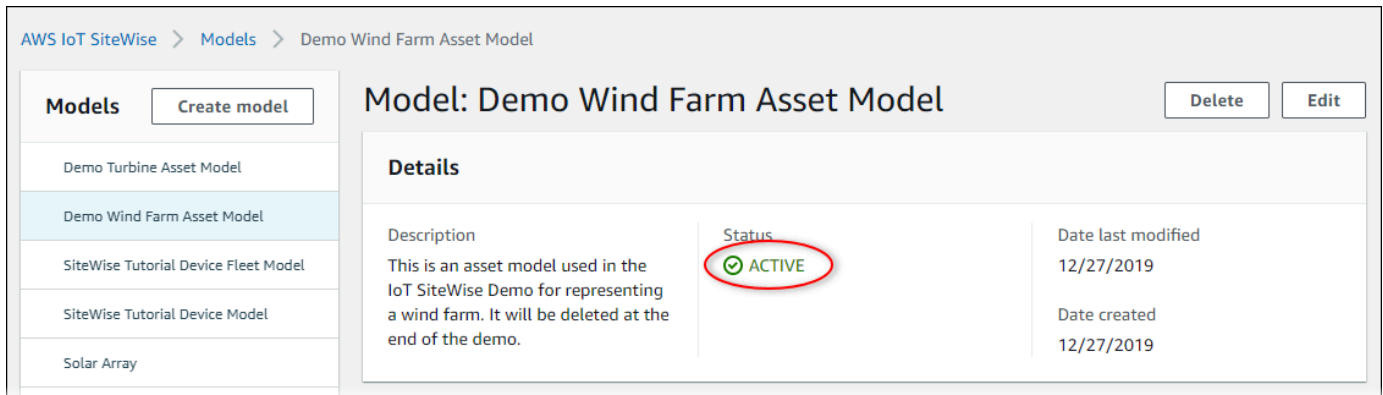
Gunakan prosedur berikut untuk memeriksa status model aset atau model komponen di AWS IoT SiteWise konsol.

Tip

Model aset dan model komponen keduanya tercantum di bawah Model di panel navigasi. Panel Detail dari model aset atau model komponen yang dipilih menunjukkan jenisnya.

Untuk memeriksa status model aset atau model komponen (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model yang akan diperiksa.
4. Temukan Status di panel Detail.



Periksa status model aset atau model komponen (AWS CLI)

Anda dapat menggunakan AWS CLI untuk memeriksa status model aset atau model komponen.

Untuk memeriksa status model aset atau model komponen, gunakan [DescribeAssetModel](#) operasi dengan `assetModelId` parameter.

Tip

AWS CLI Mendefinisikan model komponen sebagai jenis model aset. Oleh karena itu, Anda menggunakan [DescribeAssetModel](#) operasi yang sama untuk kedua jenis model. `assetModelTypeBidang` dalam respons menunjukkan apakah itu an `ASSET_MODEL` atau `aCOMPONENT_MODEL`.

Untuk memeriksa status model aset atau model komponen (AWS CLI)

- Jalankan perintah berikut untuk menggambarkan model. Ganti `asset-model-id` dengan ID atau ID eksternal model aset atau model komponen. ID eksternal adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.


```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Operasi mengembalikan respons yang berisi detail model. Respons berisi `assetModelStatus` objek yang memiliki struktur berikut.

```
{
  ...
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

Status model ada `assetModelStatus.state` di objek JSON.

Versi model aset

AWS IoT SiteWise mendukung pemrosesan asinkron untuk membuat dan memperbarui operasi pada model aset dan model komponen. Ini juga memperbarui status model.

AWS IoT SiteWise menyebarkan perubahan model yang valid dalam membuat dan memperbarui permintaan ke sumber daya dependennya (dari model aset ke aset, atau dari model komponen ke model aset). Kemudian menempatkan model di ACTIVE negara bagian.

Jika definisi model yang diberikan tidak valid, AWS IoT SiteWise tempatkan model dalam keadaan. FAILED Perubahan tidak disebarkan ke sumber daya dependen. Sumber daya dependen mengacu pada definisi model terakhir yang disebarkan ketika model berada dalam ACTIVE keadaan.

Berdasarkan informasi di atas, definisi model memiliki dua jenis versi model:

1. Versi terbaru — Definisi terbaru diterima sebagai bagian dari permintaan membuat atau memperbarui.
2. Versi aktif — Definisi terbaru berhasil diproses, dan status model adalah ACTIVE.

Secara default, detail versi terbaru model dikembalikan ketika describe APIs dipanggil pada model aset atau model komponen. Ada skenario di mana versi aktif dari model aset atau model komponen diperlukan. Lihat contoh skenario di bawah ini:

- Operasi pembaruan dengan definisi yang tidak valid menempatkan model aset Anda dalam keadaan. FAILED Anda harus mengembalikan perubahan dengan mengambil versi aktif model aset, dan membuat permintaan pembaruan lain yang mengacu pada definisi yang valid ini.
- Aplikasi AWS IoT SiteWise ada di mana pelanggan dapat melihat aset dan model aset yang sesuai. Ketika pengguna merujuk definisi model aset yang sesuai dengan aset tertentu, dan model aset dalam FAILED keadaan sementara UPDATINGPROPAGATING, versi terbaru mengembalikan definisi model aset yang belum disebar ke asetnya. Dalam hal ini, Anda harus mengambil versi aktif dari model aset kepada pelanggan.

Topik

- [Mengambil versi aktif dari model aset atau model komponen \(konsol\)](#)
- [Mengambil versi aktif dari model aset atau model komponen \(\)AWS CLI](#)

Mengambil versi aktif dari model aset atau model komponen (konsol)

Ikuti prosedur ini untuk mengambil versi aktif model aset atau model komponen di AWS IoT SiteWise konsol.

Tip

Model aset dan model komponen keduanya tercantum di bawah Model di panel navigasi. Panel Detail dari model aset atau model komponen yang dipilih menunjukkan jenisnya.

Untuk mengambil versi aktif dari model aset atau model komponen (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model untuk mengambil versi aktifnya.
 - a. Jika model dalam ACTIVE keadaan, Anda melihat versi aktifnya.

- b. Jika model dalam status sementaraUPDATING, atau FAILED statusPROPAGATING, temukan Lihat versi aktif di bawah Status di panel Detail.

Mengambil versi aktif dari model aset atau model komponen ()AWS CLI

Gunakan AWS CLI untuk mengambil versi aktif dari model aset atau model komponen.

Untuk mengambil versi aktif dari model aset atau model komponen, gunakan [DescribeAssetModel](#) operasi dengan `assetModelVersion` parameter.

Tip

AWS CLI Mendefinisikan model komponen sebagai jenis model aset. Oleh karena itu, Anda menggunakan [DescribeAssetModel](#) operasi yang sama untuk kedua jenis model. `assetModelTypeBidang` dalam respons menunjukkan apakah itu an `ASSET_MODEL` atau `aCOMPONENT_MODEL`.

Untuk mengambil versi aktif dari model aset atau model komponen ()AWS CLI

- Jalankan perintah berikut untuk menggambarkan model. Ganti *asset-model-id* dengan ID atau ID eksternal model aset atau model komponen. ID eksternal adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

Operasi mengembalikan respons dengan detail model. Respons berisi `assetModelStatus` objek dengan struktur berikut.

```
{
  ...
  "assetModelName": "string",
  "assetModelProperties": [ ... ],
  ...,
  "assetModelVersion": "string"
}
```

Model komposit khusus (komponen)

Ketika Anda memodelkan aset industri yang sangat kompleks, seperti mesin rumit yang memiliki banyak bagian, dapat menjadi tantangan untuk menjaga model aset Anda tetap teratur dan dapat dipelihara.

Dalam kasus seperti itu, Anda dapat menambahkan model komposit khusus, atau komponen jika Anda menggunakan konsol, ke model aset dan model komponen yang ada. Ini membantu Anda tetap terorganisir dengan mengelompokkan properti terkait dan menggunakan kembali definisi subkomponen.

Ada dua jenis model komposit khusus:

- Model komposit kustom sebaris mendefinisikan satu set properti yang dikelompokkan yang berlaku untuk model aset atau model komponen yang menjadi milik model komposit kustom. Anda menggunakannya untuk mengelompokkan properti terkait. Mereka terdiri dari nama, deskripsi, dan satu set properti model aset. Mereka tidak dapat digunakan kembali.
- Model komposit component-model-based kustom C mereferensikan model komponen yang ingin Anda sertakan dalam model aset atau model komponen Anda. Anda menggunakannya untuk memasukkan subrakitan standar dalam model Anda. Mereka terdiri dari nama, deskripsi, dan ID dari model komponen yang direferensikannya. Mereka tidak memiliki properti mereka sendiri; model komponen yang direferensikan menyediakan properti terkait untuk setiap aset yang dibuat.

Bagian berikut menggambarkan cara menggunakan model komposit khusus dalam desain Anda.

Topik

- [Model komposit kustom sebaris](#)
- [Component-model-based model komposit kustom](#)
- [Gunakan jalur untuk mereferensikan properti model komposit khusus](#)

Model komposit kustom sebaris

Model komposit kustom sebaris menyediakan cara untuk mengatur model aset Anda dengan mengelompokkan properti terkait.

Misalnya, Anda ingin memodelkan aset robot. Robot termasuk servomotor, catu daya, dan baterai. Masing-masing bagian penyusun memiliki sifat sendiri yang ingin Anda sertakan dalam model. Anda

dapat mendefinisikan model aset `robot_model` yang disebut yang memiliki properti seperti berikut ini.

- `robot_model`
 - `servo_status` (bilangan bulat)
 - `servo_position` (ganda)
 - `powersupply_status` (bilangan bulat)
 - `powersupply_temperature` (ganda)
 - `battery_status` (bilangan bulat)
 - `battery_charge` (ganda)

Namun, dalam beberapa kasus, mungkin ada banyak subassemblies, atau subassemblies itu sendiri mungkin memiliki banyak properti. Dalam kasus ini, mungkin ada begitu banyak properti sehingga menjadi rumit untuk direferensikan dan dipertahankan dalam satu daftar datar di root model, seperti pada contoh sebelumnya.

Untuk menghadapi situasi seperti itu, Anda dapat menggunakan model komposit kustom sebaris untuk mengelompokkan properti. Model komposit kustom inline adalah model komposit khusus yang mendefinisikan propertinya sendiri. Misalnya, Anda dapat memodelkan robot Anda seperti berikut ini.

- `robot_model`
 - `servo`
 - `status`(bilangan bulat)
 - `position`(ganda)
 - `powersupply`
 - `status`(bilangan bulat)
 - `temperature` (ganda)
 - `battery`
 - `status`(bilangan bulat)
 - `charge`(ganda)

Dalam contoh sebelumnya,, `servopowersupply`, dan `battery` adalah nama-nama model komposit kustom inline yang didefinisikan dalam model aset. `robot_model` Masing-masing model komposit ini kemudian mendefinisikan sifat-sifatnya sendiri.

Note

Dalam hal ini, setiap model komposit khusus mendefinisikan propertinya sendiri, sehingga semua properti adalah bagian dari model aset itu sendiri (`robot_model` dalam hal ini). Properti ini tidak dibagikan dengan model aset atau model komponen lainnya. Misalnya, jika Anda membuat beberapa model aset lain yang juga memiliki model komposit kustom sebaris yang disebut `servo`, maka membuat perubahan ke `servo` dalam `robot_model` tidak akan memengaruhi `servo` definisi model aset lainnya.

Jika Anda ingin menerapkan berbagi seperti itu (misalnya, hanya memiliki satu definisi untuk `servo`, yang dapat dibagikan oleh semua model aset Anda), Anda akan membuat model komponen untuknya, dan kemudian membuat model `component-model-based` komposit yang mereferensikannya. Lihat bagian berikut untuk detailnya.

Untuk informasi tentang cara membuat model komposit kustom inline, lihat [Buat model komposit khusus \(komponen\)](#).

Component-model-based model komposit kustom

Anda dapat membuat model komponen AWS IoT SiteWise untuk menentukan sub-rakitan standar yang dapat digunakan kembali. Setelah Anda membuat model komponen, Anda dapat menambahkan referensi ke model aset dan model komponen Anda yang lain. Anda melakukan ini dengan menambahkan model komposit `component-model-based` khusus ke model mana pun di mana Anda ingin mereferensikan komponen. Anda dapat menambahkan referensi ke komponen Anda dari banyak model, atau beberapa kali dalam model yang sama.

Dengan cara ini, Anda dapat menghindari duplikasi definisi yang sama di seluruh model. Ini juga menyederhanakan pemeliharaan model Anda, karena setiap perubahan yang Anda buat pada model komponen akan tercermin di semua model aset yang menggunakannya.

Misalnya, misalkan instalasi industri Anda memiliki banyak jenis peralatan yang semuanya menggunakan jenis motor servo yang sama. Beberapa dari mereka memiliki banyak motor servo dalam satu peralatan. Anda membuat model aset untuk setiap jenis peralatan, tetapi Anda tidak ingin menduplikasi definisi `servo` setiap saat. Anda ingin memodelkannya sekali saja dan

menggunakannya dalam berbagai model aset Anda. Jika nanti Anda membuat perubahan pada definisi `servo`, itu akan diperbarui di semua model dan aset Anda.

Untuk memodelkan robot dari contoh sebelumnya dengan cara ini, Anda dapat mendefinisikan motor servo, catu daya, dan baterai sebagai model komponen, seperti ini.

- `servo_component_model`
 - `status`(bilangan bulat)
 - `position`(ganda)

- `powersupply_component_model`
 - `status`(bilangan bulat)
 - `temperature` (ganda)

- `battery__component_model`
 - `status`(bilangan bulat)
 - `charge`(ganda)

Anda kemudian dapat mendefinisikan model aset, seperti `robot_model`, yang mereferensikan komponen-komponen ini. Beberapa model aset dapat mereferensikan model komponen yang sama. Anda juga dapat mereferensikan model komponen yang sama beberapa kali dalam satu model aset, seperti jika robot Anda memiliki beberapa servomotor di dalamnya.

- `robot_model`
 - `servo1`(referensi:`servo_component_model`)
 - `servo2`(referensi:`servo_component_model`)
 - `servo3`(referensi:`servo_component_model`)
 - `powersupply` (referensi:`powersupply_component_model`)
 - `battery`(referensi:`battery_component_model`)

Untuk informasi tentang cara membuat model komponen, lihat [Buat model komponen](#).

Untuk informasi tentang cara mereferensikan model komponen Anda di model lain, lihat [Buat model komposit khusus \(komponen\)](#).

Gunakan jalur untuk mereferensikan properti model komposit khusus

Saat membuat properti pada model aset, model komponen, atau model komposit kustom, Anda dapat mereferensikannya dari properti lain yang menggunakan nilainya, seperti [transformasi](#) dan [metrik](#).

AWS IoT SiteWise menyediakan berbagai cara bagi Anda untuk mereferensikan properti Anda. Cara paling sederhana adalah sering menggunakan ID propertinya. Namun, jika properti yang ingin Anda referensikan ada pada model komposit khusus, Anda mungkin merasa lebih berguna untuk mereferensikannya berdasarkan jalur.

Path adalah urutan urutan segmen jalur yang menentukan properti dalam hal posisinya di antara model komposit bersarang dalam model aset dan model komposit.

Dapatkan jalur properti

Anda bisa mendapatkan jalur properti dari path bidangnya [AssetModelProperty](#).

Misalnya, Anda memiliki model aset `robot_model` yang berisi model komposit khusus `servo`, yang memiliki properti `position`. Jika Anda memanggil [DescribeAssetModelCompositeModel](#) `servo`, maka `position` properti akan mencantumkan path bidang yang terlihat seperti ini:

```
"path": [  
  {  
    "id": "asset model ID",  
    "name": "robot_model"  
  },  
  {  
    "id": "composite model ID",  
    "name": "servo"  
  },  
  {  
    "id": "property ID",  
    "name": "position"  
  }  
]
```


Menggunakan jalur properti

Anda dapat menggunakan jalur properti saat menentukan properti yang mereferensikan properti lain, seperti transformasi atau metrik.

Sebuah properti menggunakan variabel untuk referensi properti lain. Untuk informasi lebih lanjut tentang bekerja dengan variabel, lihat [Gunakan variabel dalam ekspresi rumus](#).

Saat Anda mendefinisikan variabel untuk mereferensikan properti, Anda dapat menggunakan ID properti atau jalurnya.

Untuk menentukan variabel yang menggunakan jalur properti yang direferensikan, tentukan `propertyPath` bidang nilainya.

Misalnya, untuk menentukan model aset yang memiliki metrik yang mereferensikan properti menggunakan jalur, Anda dapat meneruskan payload seperti ini ke [CreateAssetModel](#):

```
{
  ...
  "assetModelProperties": [
    {
      ...
      "type": {
        "metric": {
          ...
          "variables": [
            {
              "name": "variable name",
              "value": {
                "propertyPath": [
                  path segments
                ]
              }
            }
          ]
        }
      },
      ...
    },
    ...
  ],
  ...
}
```

```
}
```

Mengatur AWS IoT SiteWise objek IDs

AWS IoT SiteWise mendefinisikan berbagai jenis objek persisten, seperti aset, model aset, properti, dan hierarki. Semua objek tersebut memiliki pengidentifikasi unik yang dapat Anda gunakan untuk mengambil, memperbarui, dan menghapusnya.

AWS IoT SiteWise memiliki opsi berbeda untuk pelanggan untuk pembuatan ID. AWS IoT SiteWise menghasilkan satu untuk Anda secara default pada waktu pembuatan objek. Pengguna juga dapat memberikan milik mereka IDs ke objek Anda.

Topik

- [Bekerja dengan objek UUIDs](#)
- [Gunakan eksternal IDs](#)

Bekerja dengan objek UUIDs

Setiap objek persisten AWS IoT SiteWise memiliki [UUID](#) untuk mengidentifikasinya. Misalnya, model aset memiliki ID model aset, aset memiliki ID aset, dan sebagainya. ID ini ditetapkan pada saat Anda membuat objek, dan tetap tidak berubah untuk masa pakai objek.

Saat Anda membuat objek baru, AWS IoT SiteWise buat ID unik untuk Anda secara default. Anda juga dapat memberikan ID Anda sendiri pada waktu pembuatan dalam format UUID.

Note

UUIDs harus unik secara global dalam AWS Wilayah tempat ia dibuat, dan untuk jenis objek yang sama. Ketika AWS IoT SiteWise otomatis membuat ID untuk Anda, itu selalu unik. Jika Anda memilih ID Anda sendiri, pastikan itu unik.

Misalnya, jika Anda membuat model aset baru dengan menelepon [CreateAssetModel](#), Anda dapat memberikan UUID Anda sendiri di `assetModelId` bidang opsional permintaan.

Sebaliknya, jika Anda menghilangkan `assetModelId` permintaan, AWS IoT SiteWise buat UUID untuk model aset baru.

Gunakan eksternal IDs

Untuk menentukan ID Anda sendiri dalam beberapa format selain UUID, Anda dapat menetapkan ID eksternal. Misalnya, Anda dapat melakukan ini jika Anda menggunakan kembali ID yang Anda gunakan dalam sistem yang tidak AWS, atau agar lebih mudah dibaca manusia. Eksternal IDs memiliki format yang lebih fleksibel. Anda dapat menggunakannya untuk mereferensikan objek Anda dalam operasi AWS IoT SiteWise API di mana Anda akan menggunakan UUID.

Seperti UUIDs, setiap ID eksternal harus unik dalam konteksnya. Misalnya, Anda tidak dapat memiliki dua model aset dengan ID eksternal yang sama. Juga, seperti UUIDs, objek hanya dapat memiliki satu ID eksternal dalam masa hidupnya, yang tidak dapat berubah.

Perbedaan antara eksternal IDs dan UUIDs

Eksternal IDs berbeda dari UUIDs dalam cara-cara berikut:

- Setiap objek memiliki UUID, tetapi eksternal IDs adalah opsional.
- AWS IoT SiteWise tidak pernah menghasilkan eksternal IDs. Anda menyediakan ini sendiri.
- Jika objek belum memilikinya, Anda dapat menetapkan ID eksternal kapan saja.

Format eksternal IDs

ID eksternal yang valid memiliki properti berikut:

- Panjangnya antara 2 dan 128 karakter.
- Karakter pertama dan terakhir harus alfanumerik (A-Z, a-z, 0-9).
- Karakter selain yang pertama dan terakhir harus alfanumerik, atau salah satu dari berikut ini: `_` `-` `.` `:`

Misalnya, ID eksternal harus sesuai dengan ekspresi reguler berikut:

```
[a-zA-Z0-9][a-zA-Z0-9_\-.:]*[a-zA-Z0-9]+
```

Referensi objek dengan eksternal IDs

Di banyak tempat di mana Anda dapat mereferensikan objek menggunakan UUID-nya, Anda dapat menggunakan ID eksternalnya sebagai gantinya, jika ada. Untuk melakukannya, tambahkan ID eksternal ke `stringexternalId`:

Misalnya, Anda memiliki model aset yang UUIDnya (ID model aset) `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`, yang juga memiliki ID eksternal. `myExternalId` Hubungi [DescribeAssetModel](#) untuk mendapatkan detail tentang hal itu. Anda dapat menggunakan salah satu dari berikut ini sebagai nilai `assetModelId`:

- Dengan ID model aset (UUID) itu sendiri: `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`
- Dengan ID eksternal: `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

`externalId`:Awalan itu sendiri bukan bagian dari ID eksternal. Anda hanya perlu memberikan awalan ketika Anda menyediakan ID eksternal ke operasi API yang menerima salah satu UUIDs atau eksternal. IDs Misalnya, berikan awalan saat Anda menanyakan atau memperbarui objek yang ada.

Saat Anda menentukan ID eksternal untuk objek, seperti saat Anda membuat model aset, jangan sertakan awalan.

Anda dapat menggunakan eksternal IDs sebagai pengganti dengan UUIDs cara ini untuk banyak operasi API di AWS IoT SiteWise, tetapi tidak semua. Misalnya, [GetAssetPropertyValue](#), harus menggunakan UUIDs; itu tidak mendukung penggunaan ID eksternal.

Untuk menentukan apakah operasi API tertentu mendukung penggunaan ini, lihat [Referensi API](#).

Buat model aset dan komponen untuk AWS IoT SiteWise

AWS IoT SiteWise model aset dan model komponen mendorong standardisasi data industri Anda. Model aset dan komponen mewakili struktur dan properti aset industri Anda dan komponennya. Model aset menentukan keseluruhan aset, seperti turbin angin atau jalur manufaktur. Model komponen mewakili komponen individu yang membentuk aset, seperti bilah, generator, atau sensor. Dengan membuat model-model ini, Anda dapat mengatur dan menyusun data aset Anda dengan cara yang mencerminkan hubungan dunia nyata dan hierarki peralatan industri Anda, sehingga lebih mudah untuk memantau, menganalisis, dan memelihara.

Model aset atau model komponen berisi nama, deskripsi, properti aset, dan (opsional) model komposit kustom yang mengelompokkan properti bersama-sama, atau model komponen referensi untuk subrakitan.

Di AWS IoT SiteWise, Anda dapat membuat model aset dan model komponen untuk mewakili struktur dan properti aset industri Anda dan komponennya.

- Anda menggunakan model aset untuk membuat aset. Selain fitur yang tercantum di atas, model aset juga dapat berisi definisi hierarki yang mendefinisikan hubungan antar aset.
- Model komponen mewakili subassembly dalam model aset atau model komponen lain. Saat membuat model komponen, Anda dapat menambahkan referensi ke model aset dan model komponen lainnya. Namun, Anda tidak dapat membuat aset langsung dari model komponen.

Setelah Anda membuat model aset atau model komponen, Anda dapat membuat model komposit khusus untuk mengelompokkan properti bersama-sama atau untuk mereferensikan model komponen yang ada.

Untuk detail tentang cara membuat model aset dan model komponen, lihat bagian berikut.

Topik

- [Buat model aset di AWS IoT SiteWise](#)
- [Buat model komponen](#)
- [Tentukan properti data](#)
- [Buat model komposit khusus \(komponen\)](#)

Buat model aset di AWS IoT SiteWise

AWS IoT SiteWise model aset mendorong standardisasi data industri Anda. Model aset berisi nama, deskripsi, properti aset, dan definisi hierarki aset. Misalnya, Anda dapat menentukan model turbin angin dengan suhu, rotasi per menit (RPM), dan sifat daya. Kemudian, Anda dapat menentukan model ladang angin dengan properti output daya bersih dan definisi hierarki turbin angin.

Note

- Kami menyarankan Anda memodelkan operasi Anda dimulai dengan node tingkat terendah. Misalnya, buat model turbin angin Anda sebelum Anda membuat model ladang

angin Anda. Definisi hierarki aset berisi referensi ke model aset yang ada. Dengan pendekatan ini, Anda dapat menentukan hierarki aset saat membuat model.

- Model aset tidak dapat berisi model aset lainnya. Jika Anda harus mendefinisikan model yang dapat Anda referensikan sebagai subassembly dalam model lain, Anda harus membuat komponen -> model sebagai gantinya. Untuk informasi selengkapnya, lihat [Buat model komponen](#).

Bagian berikut menjelaskan cara menggunakan AWS IoT SiteWise konsol atau API untuk membuat model aset. Bagian berikut juga menjelaskan berbagai jenis properti aset dan hierarki aset yang dapat Anda gunakan untuk membuat model.

Topik

- [Buat model aset \(konsol\)](#)
- [Buat model aset \(AWS CLI\)](#)
- [Contoh model aset](#)
- [Tentukan hierarki model aset](#)

Buat model aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk membuat model aset. AWS IoT SiteWise Konsol menyediakan berbagai fitur, seperti formula auto completion, yang dapat membantu Anda menentukan model aset yang valid.

Untuk membuat model aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih Buat model.
4. Pada halaman Create model, lakukan hal berikut:
 - a. Masukkan Nama untuk model aset, seperti **Wind Turbine** atau **Wind Turbine Model**. Nama ini harus unik di semua model di akun Anda di Wilayah ini.
 - b. (Opsional) Tambahkan ID Eksternal untuk model. Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

- c. (Opsional) Tambahkan definisi Pengukuran untuk model. Pengukuran mewakili aliran data dari peralatan Anda. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).
- d. (Opsional) Tambahkan definisi Transform untuk model. Transformasi adalah rumus yang memetakan data dari satu bentuk ke bentuk lainnya. Untuk informasi selengkapnya, lihat [Mengubah data \(mengubah\)](#).
- e. (Opsional) Tambahkan definisi Metrik untuk model. Metrik adalah rumus yang mengumpulkan data dari interval waktu. Metrik dapat memasukkan data dari aset terkait, sehingga Anda dapat menghitung nilai yang mewakili operasi Anda atau subset operasi Anda. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).
- f. (Opsional) Tambahkan definisi Hierarki untuk model. Hierarki adalah hubungan antar aset. Untuk informasi selengkapnya, lihat [Tentukan hierarki model aset](#).
- g. (Opsional) Tambahkan tag untuk model aset. Untuk informasi selengkapnya, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).
- h. Pilih Buat model.

Saat Anda membuat model aset, AWS IoT SiteWise konsol akan menavigasi ke halaman model baru. Pada halaman ini, Anda dapat melihat Status model, yang awalnya MENCIPTAKAN. Halaman ini diperbarui secara otomatis, sehingga Anda dapat menunggu status model diperbarui.

 Note

Proses pembuatan model aset dapat memakan waktu hingga beberapa menit untuk model yang kompleks. Setelah status model aset AKTIF, Anda dapat menggunakan model aset untuk membuat aset. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

- 5. (Opsional) Setelah membuat model aset, Anda dapat mengonfigurasi model aset untuk edge. Untuk informasi lebih lanjut tentang SiteWise Edge, lihat [Konfigurasi kemampuan tepi di AWS IoT SiteWise Edge](#).
 - a. Pada halaman model, pilih Configure for Edge.
 - b. Pada halaman konfigurasi model, pilih konfigurasi tepi untuk model Anda. Ini mengontrol tempat AWS IoT SiteWise dapat menghitung dan menyimpan properti yang terkait dengan

model aset ini. Untuk informasi selengkapnya tentang mengonfigurasi model Anda untuk edge, lihat [Siapkan sumber OPC UA di SiteWise Edge](#).

- c. Untuk konfigurasi Custom edge, pilih lokasi yang AWS IoT SiteWise ingin Anda hitung dan simpan setiap properti model aset Anda.

 Note

Transformasi dan metrik yang terkait harus dikonfigurasi untuk lokasi yang sama. Untuk informasi selengkapnya tentang mengonfigurasi model Anda untuk edge, lihat [Siapkan sumber OPC UA di SiteWise Edge](#).

- d. Pilih Simpan. Pada halaman model, konfigurasi Edge Anda sekarang harus Dikonfigurasi.

Buat model aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat model aset.

Gunakan [CreateAssetModel](#) operasi untuk membuat model aset dengan properti dan hierarki. Operasi ini mengharapkan muatan dengan struktur berikut.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
}
```

Untuk membuat model aset (AWS CLI)

1. Buat file bernama `asset-model-payload.json` dan kemudian salin objek JSON berikut ke dalam file.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ],
}
```



```
"assetModelHierarchies": [  
  
],  
"assetModelCompositeModels": [  
  
]  
}
```

2. Gunakan editor teks JSON pilihan Anda untuk mengedit `asset-model-payload.json` file sebagai berikut:
 - a. Masukkan name (`assetModelName`) untuk model aset, seperti **Wind Turbine** atau **Wind Turbine Model**. Nama ini harus unik di semua model aset dan model komponen di akun Anda dalam hal ini Wilayah AWS.
 - b. (Opsional) Masukkan ID eksternal (`assetModelExternalId`) untuk model aset. Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
 - c. (Opsional) Masukkan deskripsi (`assetModelDescription`) untuk model aset, atau hapus pasangan `assetModelDescription` nilai kunci.
 - d. (Opsional) Tentukan properti aset (`assetModelProperties`) untuk model. Untuk informasi selengkapnya, lihat [Tentukan properti data](#).
 - e. (Opsional) Tentukan hierarki aset (`assetModelHierarchies`) untuk model. Untuk informasi selengkapnya, lihat [Tentukan hierarki model aset](#).
 - f. (Opsional) Tentukan alarm untuk model. Alarm memantau properti lain sehingga Anda dapat mengidentifikasi kapan peralatan atau proses memerlukan perhatian. Setiap definisi alarm adalah model komposit (`assetModelCompositeModels`) yang menstandarisasi serangkaian properti yang digunakan alarm. Untuk informasi selengkapnya, lihat [Pantau data dengan alarm di AWS IoT SiteWise](#) dan [Tentukan alarm pada model aset di AWS IoT SiteWise](#).
 - g. (Opsional) Tambahkan tag (`tags`) untuk model aset. Untuk informasi selengkapnya, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).
3. Jalankan perintah berikut untuk membuat model aset dari definisi dalam file JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

Operasi mengembalikan respons yang berisi `assetModelId` yang Anda rujuk saat membuat aset. Respons juga berisi status model (`assetModelStatus.state`), yang awalnya `CREATING`. Status model aset adalah `CREATING` sampai perubahan menyebar.

 Note

Proses pembuatan model aset dapat memakan waktu hingga beberapa menit untuk model yang kompleks. Untuk memeriksa status model aset Anda saat ini, gunakan [DescribeAssetModel](#) operasi dengan menentukan `assetModelId`. Setelah status model aset `ACTIVE`, Anda dapat menggunakan model aset untuk membuat aset. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

4. (Opsional) Buat model komposit khusus untuk model aset Anda. Dengan model komposit khusus, Anda dapat mengelompokkan properti dalam model, atau menyertakan subassembly dengan mereferensikan model komponen. Untuk informasi selengkapnya, lihat [Buat model komposit khusus \(komponen\)](#).

Contoh model aset

Bagian ini berisi contoh definisi model aset yang dapat Anda gunakan untuk membuat model aset dengan AWS CLI dan AWS IoT SiteWise SDKs. Model aset ini mewakili turbin angin dan ladang angin. Aset turbin angin menelan data sensor mentah dan menghitung nilai seperti daya dan kecepatan angin rata-rata. Aset ladang angin menghitung nilai seperti daya total untuk semua turbin angin di ladang angin.

Topik

- [Model aset turbin angin](#)
- [Model aset peternakan angin](#)

Model aset turbin angin

Model aset berikut mewakili turbin di ladang angin. Turbin angin menelan data sensor untuk menghitung nilai seperti daya dan kecepatan angin rata-rata.

 Note

Model contoh ini menyerupai model turbin angin dari AWS IoT SiteWise demo. Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise demo](#).

```
{
  "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": []
}

```

Model aset peternakan angin

Model aset berikut mewakili ladang angin yang terdiri dari beberapa turbin angin. Model aset ini mendefinisikan [hierarki](#) model turbin angin. Hal ini memungkinkan ladang angin menghitung nilai (seperti daya rata-rata) dari data untuk semua turbin angin di ladang angin.

Note

Model contoh ini menyerupai model ladang angin dari AWS IoT SiteWise demo. Untuk informasi selengkapnya, lihat [Gunakan AWS IoT SiteWise demo](#).

Model aset ini tergantung pada [Model aset turbin angin](#). Ganti `propertyId` dan `childAssetModelId` nilai dengan yang berasal dari model aset turbin angin yang ada.

```
{
  "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"
    }
]

```

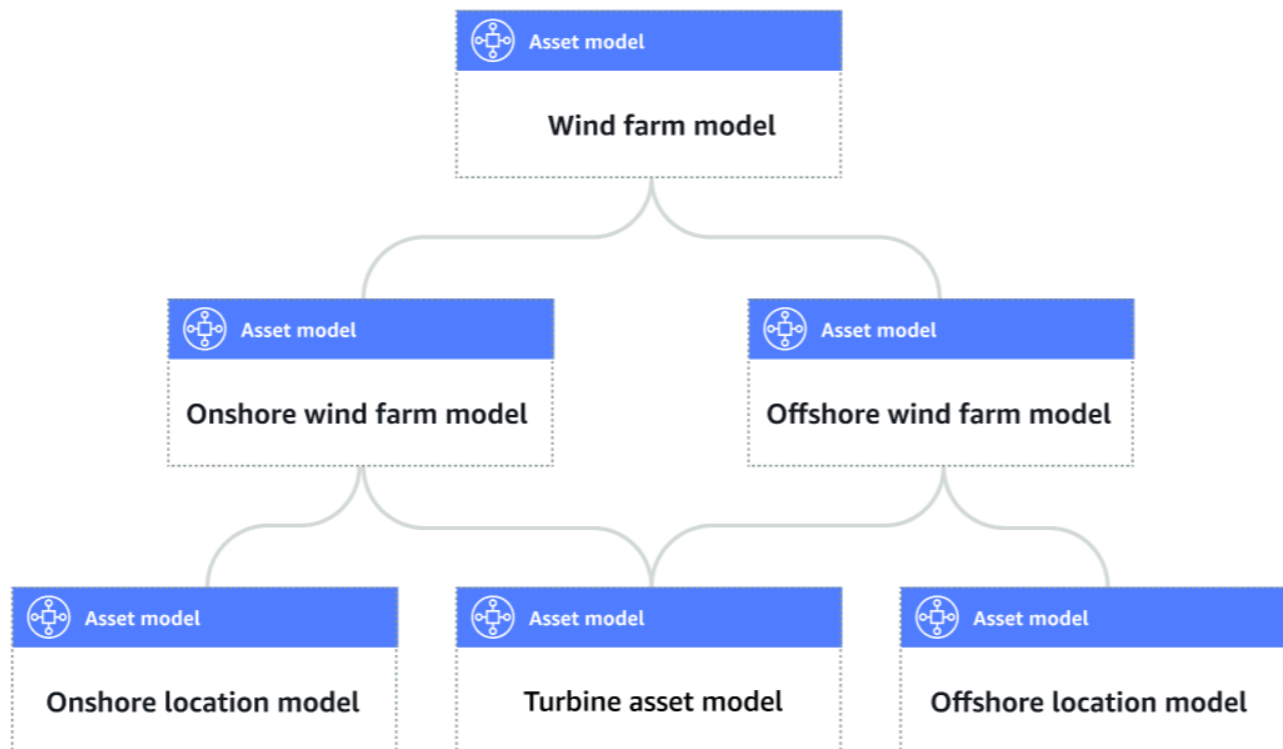
```
}  
]  
}
```

Tentukan hierarki model aset

Anda dapat menentukan hierarki model aset untuk membuat asosiasi logis antara model aset dalam operasi industri Anda. Misalnya, Anda dapat mendefinisikan ladang angin yang terdiri dari ladang angin darat dan lepas pantai. Sebuah ladang angin darat berisi turbin dan lokasi darat. Sebuah ladang angin lepas pantai berisi turbin dan lokasi lepas pantai.



Asset model hierarchy



Saat Anda mengaitkan model aset anak ke model aset induk melalui hierarki, metrik model aset induk dapat memasukkan data dari metrik model aset anak. Anda dapat menggunakan hierarki dan metrik model aset untuk menghitung statistik yang memberikan wawasan tentang operasi Anda atau

subset operasi Anda. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).

Setiap hierarki mendefinisikan hubungan antara model aset induk dan model aset anak. Dalam model aset induk, Anda dapat menentukan beberapa hierarki ke model aset anak yang sama. Misalnya, jika Anda memiliki dua jenis turbin angin yang berbeda di ladang angin Anda, di mana semua turbin angin diwakili oleh model aset yang sama, Anda dapat menentukan hierarki untuk setiap jenis. Kemudian, Anda dapat menentukan metrik dalam model ladang angin untuk menghitung statistik independen dan gabungan untuk setiap jenis turbin angin.

Model aset induk dapat dikaitkan dengan beberapa model aset anak. Misalnya, jika Anda memiliki ladang angin darat dan ladang angin lepas pantai yang diwakili oleh dua model aset yang berbeda, Anda dapat mengaitkan model aset ini dengan model aset ladang angin induk yang sama.

Model aset anak juga dapat dikaitkan dengan beberapa model aset induk. Misalnya, jika Anda memiliki dua jenis ladang angin yang berbeda, di mana semua turbin angin diwakili oleh model aset yang sama, Anda dapat mengaitkan model aset turbin angin dengan model aset ladang angin yang berbeda.

Note

Saat Anda menentukan hierarki model aset, model aset anak harus ACTIVE atau memiliki ACTIVE versi sebelumnya. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

Setelah menentukan model aset hierarkis dan membuat aset, Anda dapat mengaitkan aset untuk menyelesaikan hubungan induk-anak. Untuk informasi selengkapnya, lihat [Buat aset untuk model aset di AWS IoT SiteWise](#) dan [Mengasosiasikan dan memisahkan aset](#).

Topik

- [Tentukan hierarki model aset \(konsol\)](#)
- [Tentukan hierarki aset \(AWS CLI\)](#)

Tentukan hierarki model aset (konsol)

Saat Anda menentukan hierarki untuk model aset di AWS IoT SiteWise konsol, Anda menentukan parameter berikut:

- Nama hierarki — Nama hierarki, seperti. **Wind Turbines**

- Model hierarki — Model aset anak.
- Hierarki ID Eksternal (Opsional) - Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Untuk informasi selengkapnya, lihat [Buat model aset \(konsol\)](#).

Tentukan hierarki aset ()AWS CLI

Saat Anda menentukan hierarki untuk model aset dengan AWS IoT SiteWise API, Anda menentukan parameter berikut:

- name— Nama hierarki, seperti **Wind Turbines**.
- childAssetModelId— ID atau ID eksternal model aset anak untuk hierarki. Anda dapat menggunakan [ListAssetModels](#) operasi untuk menemukan ID model aset yang ada.

Example Contoh definisi hierarki

Contoh berikut menunjukkan hierarki model aset yang mewakili hubungan ladang angin dengan turbin angin. Objek ini adalah contoh dari sebuah [AssetModelHierarchy](#). Untuk informasi selengkapnya, lihat [Buat model aset \(AWS CLI\)](#).

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    },
  ]
}
```

Buat model komponen

Gunakan model AWS IoT SiteWise komponen untuk menentukan subrakitan yang dapat Anda referensikan dari model aset atau model komponen lainnya. Dengan cara ini, Anda dapat menggunakan kembali definisi komponen di beberapa model lain, atau beberapa kali dalam model yang sama.

Proses mendefinisikan model komponen sangat mirip dengan mendefinisikan model aset. Seperti model aset, model komponen memiliki nama, deskripsi, dan properti aset. Namun, model komponen tidak dapat menyertakan definisi hierarki aset, karena model komponen itu sendiri tidak dapat digunakan untuk membuat aset secara langsung. Model komponen juga tidak dapat mendefinisikan alarm.

Misalnya, Anda dapat menentukan komponen untuk motor servo dengan suhu motor, suhu encoder, dan sifat resistansi isolasi. Kemudian, Anda dapat menentukan model aset untuk peralatan yang berisi motor servo, seperti mesin CNC.

Note

- Kami menyarankan Anda memodelkan operasi Anda dimulai dengan node tingkat terendah. Misalnya, buat komponen motor servo Anda sebelum Anda membuat model aset mesin CNC Anda. Model aset berisi referensi ke model komponen yang ada.
- Anda tidak dapat membuat aset langsung dari model komponen. Untuk membuat aset yang menggunakan komponen Anda, Anda harus membuat model aset untuk aset Anda. Kemudian, Anda membuat model komposit khusus untuk itu yang mereferensikan komponen Anda. Untuk informasi selengkapnya tentang membuat model aset, lihat [Buat model aset di AWS IoT SiteWise](#) Untuk informasi selengkapnya tentang membuat model komposit kustom, lihat [Buat model komposit khusus \(komponen\)](#).

Bagian berikut menjelaskan cara menggunakan AWS IoT SiteWise API untuk membuat model komponen.

Topik

- [Buat model komponen \(AWS CLI\)](#)
- [Contoh model komponen](#)

Buat model komponen (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat model komponen.

Gunakan [CreateAssetModel](#) operasi untuk membuat model komponen dengan properti. Operasi ini mengharapkan muatan dengan struktur berikut:

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
}
```

Untuk membuat model komponen (AWS CLI)

1. Buat file bernama `component-model-payload.json` dan kemudian salin objek JSON berikut ke dalam file:

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ]
}
```

2. Gunakan editor teks JSON pilihan Anda untuk mengedit `component-model-payload.json` file sebagai berikut:
 - a. Masukkan nama (`assetModelName`) untuk model komponen, seperti **Servo Motor** atau **Servo Motor Model**. Nama ini harus unik di semua model aset dan model komponen di akun Anda dalam hal ini Wilayah AWS.
 - b. (Opsional) Masukkan ID eksternal (`assetModelExternalId`) untuk model komponen. Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
 - c. (Opsional) Masukkan deskripsi (`assetModelDescription`) untuk model aset, atau hapus pasangan `assetModelDescription` nilai kunci.
 - d. (Opsional) Tentukan properti aset (`assetModelProperties`) untuk model komponen. Untuk informasi selengkapnya, lihat [Tentukan properti data](#).
 - e. (Opsional) Tambahkan tag (`tags`) untuk model aset. Untuk informasi selengkapnya, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).
3. Jalankan perintah berikut untuk membuat model komponen dari definisi dalam file JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-payload.json
```

Operasi mengembalikan respons yang berisi `assetModelId` yang Anda rujuk saat menambahkan referensi ke model komponen Anda dalam model aset atau model komponen lain. Respons juga berisi status model (`assetModelStatus.state`), yang awalnya `CREATING`. Status model komponen adalah `CREATING` sampai perubahan menyebar.

Note

Proses pembuatan model komponen dapat memakan waktu hingga beberapa menit untuk model yang kompleks. Untuk memeriksa status model komponen Anda saat ini, gunakan [DescribeAssetModel](#) operasi dengan menentukan `assetModelId`. Setelah status model komponen `ACTIVE`, Anda dapat menambahkan referensi ke model komponen Anda dalam model aset atau model komponen lainnya. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

4. (Opsional) Buat model komposit khusus untuk model komponen Anda. Dengan model komposit khusus, Anda dapat mengelompokkan properti dalam model, atau menyertakan subassembly dengan mereferensikan model komponen lain. Untuk informasi selengkapnya, lihat [Buat model komposit khusus \(komponen\)](#).

Contoh model komponen

Bagian ini berisi contoh definisi model komponen yang dapat Anda gunakan untuk membuat model komponen dengan AWS CLI dan AWS IoT SiteWise SDKs. Model komponen ini mewakili motor servo yang dapat digunakan dalam peralatan lain, seperti mesin CNC.

Topik

- [Model komponen motor servo](#)

Model komponen motor servo

Model komponen berikut mewakili motor servo yang dapat digunakan dalam peralatan seperti mesin CNC. Motor servo menyediakan berbagai pengukuran, seperti suhu dan hambatan listrik.

Pengukuran ini tersedia sebagai properti pada aset yang dibuat dari model aset yang merujuk pada model komponen motor servo.

```
{
  "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"
    }
  ]
}
```

Tentukan properti data

Properti aset adalah struktur dalam setiap aset yang berisi data aset. Properti aset dapat berupa salah satu dari jenis berikut:

- **Atribut** — Properti aset yang umumnya statis, seperti produsen perangkat atau wilayah geografis. Untuk informasi selengkapnya, lihat [Tentukan data statis \(atribut\)](#).
- **Pengukuran** — Aliran data sensor perangkat mentah aset, seperti nilai kecepatan rotasi stempel waktu atau nilai suhu stempel waktu dalam Celcius. Pengukuran didefinisikan oleh alias aliran data. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).
- **Transformasi** — Nilai deret waktu yang ditransformasikan aset, seperti nilai suhu stempel waktu di Fahrenheit. Transformasi didefinisikan oleh ekspresi dan variabel untuk dikonsumsi dengan ekspresi itu. Untuk informasi selengkapnya, lihat [Mengubah data \(mengubah\)](#).

- **Metrik** — Data aset yang dikumpulkan selama interval waktu tertentu, seperti suhu rata-rata per jam. Metrik didefinisikan oleh interval waktu, ekspresi, dan variabel untuk dikonsumsi dengan ekspresi itu. Ekspresi metrik dapat memasukkan properti metrik aset terkait, sehingga Anda dapat menghitung metrik yang mewakili operasi Anda atau subset operasi Anda. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).

Untuk informasi selengkapnya, lihat [Buat model aset di AWS IoT SiteWise](#).

Untuk contoh cara menggunakan pengukuran, transformasi, dan metrik untuk menghitung Overall Equipment Effectivity (OEE), lihat. [Hitung OEE di AWS IoT SiteWise](#)

Topik

- [Tentukan data statis \(atribut\)](#)
- [Tentukan aliran data dari peralatan \(pengukuran\)](#)
- [Mengubah data \(mengubah\)](#)
- [Data agregat dari properti dan aset lainnya \(metrik\)](#)
- [Gunakan ekspresi rumus](#)

Tentukan data statis (atribut)

Atribut aset mewakili informasi yang umumnya statis, seperti produsen perangkat atau lokasi geografis. Setiap aset yang Anda buat dari model aset berisi atribut model tersebut.

Topik

- [Tentukan atribut \(konsol\)](#)
- [Tentukan atribut \(AWS CLI\)](#)

Tentukan atribut (konsol)

Saat menentukan atribut untuk model aset di AWS IoT SiteWise konsol, Anda menentukan parameter berikut:

- **Nama** - Nama properti.
- **Nilai default** — (Opsional) Nilai default untuk atribut ini. Aset yang dibuat dari model memiliki nilai ini untuk atribut. Untuk informasi selengkapnya tentang cara mengganti nilai default dalam aset yang dibuat dari model, lihat [Perbarui nilai atribut](#).

- Tipe data — Tipe data properti, yang merupakan salah satu dari berikut ini:
 - String — Sebuah string dengan hingga 1024 byte.
 - Integer — Bilangan bulat 32-bit yang ditandatangani dengan rentang [-2,147.483.648, 2,147.483.647].
 - Ganda — Nomor floating point dengan jangkauan [-10¹⁰⁰, 10¹⁰⁰] dan presisi ganda IEEE 754.
 - Boolean — true atau false.
- ID Eksternal — (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Untuk informasi selengkapnya, lihat [Buat model aset \(konsol\)](#).

Tentukan atribut (AWS CLI)

Saat menentukan atribut untuk model aset dengan AWS IoT SiteWise API, Anda menentukan parameter berikut:

- name- Nama properti.
- defaultValue— (Opsional) Nilai default untuk atribut ini. Aset yang dibuat dari model memiliki nilai ini untuk atribut. Untuk informasi selengkapnya tentang cara mengganti nilai default dalam aset yang dibuat dari model, lihat [Perbarui nilai atribut](#).
- dataType— Tipe data properti, yang merupakan salah satu dari berikut ini:
 - STRING— Sebuah string dengan hingga 1024 byte.
 - INTEGER— Bilangan bulat 32-bit yang ditandatangani dengan rentang [-2,147.483.648, 2.147.483.647].
 - DOUBLE— Nomor floating point dengan jangkauan [-10¹⁰⁰, 10¹⁰⁰] dan presisi ganda IEEE 754.
 - BOOLEAN— true atau false.
- externalId— (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Example Contoh definisi atribut

Contoh berikut menunjukkan atribut yang mewakili nomor model aset dengan nilai default. Objek ini adalah contoh dari sebuah [AssetModelProperty](#) yang berisi [Atribut](#). Anda dapat menentukan objek ini sebagai bagian dari payload [CreateAssetModel](#) permintaan untuk membuat properti atribut. Untuk informasi selengkapnya, lihat [Buat model aset \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

Tentukan aliran data dari peralatan (pengukuran)

Pengukuran mewakili aliran data sensor mentah perangkat, seperti nilai suhu stempel waktu atau nilai rotasi per menit (RPM) stempel waktu.

Topik

- [Tentukan pengukuran \(konsol\)](#)
- [Tentukan pengukuran \(AWS CLI\)](#)

Tentukan pengukuran (konsol)

Saat menentukan pengukuran untuk model aset di AWS IoT SiteWise konsol, Anda menentukan parameter berikut:

- Nama - Nama properti.
- Unit — (Opsional) Unit ilmiah untuk properti, seperti mm atau Celcius.
- Tipe data — Tipe data properti, yang merupakan salah satu dari berikut ini:

- String — Sebuah string dengan hingga 1024 byte.
- Integer — Bilangan bulat 32-bit yang ditandatangani dengan rentang [-2,147.483.648, 2,147.483.647].
- Ganda — Nomor floating point dengan jangkauan [-10¹⁰⁰, 10¹⁰⁰] dan presisi ganda IEEE 754.
- Boolean — `true` atau `false`.
- ID Eksternal — (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Untuk informasi selengkapnya, lihat [Buat model aset \(konsol\)](#).

Tentukan pengukuran (AWS CLI)

Saat menentukan pengukuran untuk model aset dengan AWS IoT SiteWise API, Anda menentukan parameter berikut:

- `name`- Nama properti.
- `dataType`— Tipe data properti, yang merupakan salah satu dari berikut ini:
 - `STRING`— Sebuah string dengan hingga 1024 byte.
 - `INTEGER`— Bilangan bulat 32-bit yang ditandatangani dengan rentang [-2,147.483.648, 2.147.483.647].
 - `DOUBLE`— Nomor floating point dengan jangkauan [-10¹⁰⁰, 10¹⁰⁰] dan presisi ganda IEEE 754.
 - `BOOLEAN`— `true` atau `false`.
- `unit`— (Opsional) Unit ilmiah untuk properti, seperti mm atau Celcius.
- `externalId`— (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Example Contoh definisi pengukuran

Contoh berikut menunjukkan pengukuran yang mewakili pembacaan sensor suhu aset. Objek ini adalah contoh dari sebuah [AssetModelProperty](#) yang berisi [Pengukuran](#). Anda dapat menentukan objek ini sebagai bagian dari payload [CreateAssetModel](#) permintaan untuk membuat properti pengukuran. Untuk informasi selengkapnya, lihat [Buat model aset \(AWS CLI\)](#).

Struktur [Pengukuran](#) adalah struktur kosong saat Anda menentukan model aset karena nanti Anda mengonfigurasi setiap aset untuk menggunakan aliran data perangkat yang unik. Untuk informasi selengkapnya tentang cara menghubungkan properti pengukuran aset ke aliran data sensor perangkat, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    }
  ],
  ...
}
```

Mengubah data (mengubah)

Transformasi adalah ekspresi matematika yang memetakan titik data properti aset dari satu bentuk ke bentuk lainnya. Ekspresi transformasi terdiri dari variabel properti aset, literal, operator, dan fungsi. Titik data yang diubah memiliki one-to-one hubungan dengan titik data input. AWS IoT SiteWise menghitung titik data baru yang diubah setiap kali salah satu properti input menerima titik data baru.

Note

Untuk pembaruan properti dengan stempel waktu yang sama, nilai keluaran dapat ditimpa oleh pembaruan dari properti masuk lainnya.

Misalnya, jika aset Anda memiliki aliran pengukuran suhu yang diberi nama `Temperature_C` dengan satuan dalam Celcius, Anda dapat mengonversi setiap titik data menjadi Fahrenheit dengan rumus $\text{Temperature\_F} = 9/5 * \text{Temperature\_C} + 32$. Setiap kali AWS IoT SiteWise menerima titik data dalam aliran `Temperature_C` pengukuran, `Temperature_F` nilai yang sesuai dihitung dalam beberapa detik dan tersedia sebagai `Temperature_F` properti.

Jika transformasi Anda berisi lebih dari satu variabel, titik data yang tiba lebih awal segera memulai perhitungan. Pertimbangkan contoh di mana produsen suku cadang menggunakan transformasi untuk memantau kualitas produk. Menggunakan standar yang berbeda berdasarkan jenis bagian, pabrikan menggunakan pengukuran berikut untuk mewakili proses:

- **Part_Number**- String yang mengidentifikasi tipe bagian.
- **Good_Count**- Bilangan bulat yang bertambah satu jika bagian tersebut memenuhi standar.
- **Bad_Count**- Bilangan bulat yang bertambah satu jika bagian tidak memenuhi standar.

Pabrikan juga menciptakan transformasi, **Quality_Monitor**, yang sama. `if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")`

Transformasi ini memantau persentase bagian buruk yang diproduksi untuk jenis bagian tertentu. Jika nomor bagiannya BLT123 dan persentase bagian yang buruk melebihi 10 persen (0,1), transformasi kembali "Caution". Jika tidak, transformasi akan kembali "Normal".

Note

- Jika **Part_Number** menerima titik data baru sebelum pengukuran lain, **Quality_Monitor** transformasi menggunakan **Part_Number** nilai baru **Good_Count** dan **Bad_Count** nilai terbaru. Untuk menghindari kesalahan, setel ulang **Good_Count** dan **Bad_Count** sebelum pembuatan berikutnya dijalankan.
- Gunakan [metrik](#) jika Anda ingin mengevaluasi ekspresi hanya setelah semua variabel menerima titik data baru.

Topik

- [Tentukan transformasi \(konsol\)](#)
- [Tentukan transformasi \(\)AWS CLI](#)

Tentukan transformasi (konsol)

Saat menentukan transformasi untuk model aset di AWS IoT SiteWise konsol, Anda menentukan parameter berikut:

- Nama - Nama properti.
- Unit — (Opsional) Unit ilmiah untuk properti, seperti mm atau Celcius.
- Tipe data — Tipe data dari transformasi, yang dapat berupa Double atau String.
- ID Eksternal — (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
- Formula - Ekspresi transformasi. Ekspresi transformasi tidak dapat menggunakan fungsi agregasi atau fungsi temporal. Untuk membuka fitur auto complete, mulai mengetik atau tekan tombol panah bawah. Untuk informasi selengkapnya, lihat [Gunakan ekspresi rumus](#).

Important

Transformasi dapat memasukkan properti yang integer, ganda, Boolean, atau tipe string. Boolean mengkonversi ke 0 (false) dan 1 (true).

Transformasi harus memasukkan satu atau lebih properti yang bukan atribut dan sejumlah properti atribut. AWS IoT SiteWise menghitung titik data baru yang diubah setiap kali properti input yang bukan atribut menerima titik data baru. Nilai atribut baru tidak meluncurkan pembaruan transformasi. Tingkat permintaan yang sama untuk operasi API data properti aset berlaku untuk hasil komputasi transformasi.

Ekspresi rumus hanya dapat menampilkan nilai ganda atau string. Ekspresi bersarang dapat menampilkan tipe data lain, seperti string, tetapi rumus secara keseluruhan harus mengevaluasi ke angka atau string. Anda dapat menggunakan [fungsi jp](#) untuk mengonversi string ke angka. Nilai Boolean harus 1 (true) atau 0 (false). Untuk informasi selengkapnya, lihat [Nilai tidak terdefinisi, tak terbatas, dan meluap](#).

Untuk informasi selengkapnya, lihat [Buat model aset \(konsol\)](#).

Tentukan transformasi ()AWS CLI

Saat menentukan transformasi untuk model aset dengan AWS IoT SiteWise API, Anda menentukan parameter berikut:

- name- Nama properti.
- unit— (Opsional) Unit ilmiah untuk properti, seperti mm atau Celcius.
- dataType— Tipe data dari transformasi, yang harus DOUBLE atauSTRING.

- **externalId**— (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
- **expression**Ekspresi transformasi. Ekspresi transformasi tidak dapat menggunakan fungsi agregasi atau fungsi temporal. Untuk informasi selengkapnya, lihat [Gunakan ekspresi rumus](#).
- **variables**— Daftar variabel yang mendefinisikan properti lain dari aset Anda untuk digunakan dalam ekspresi. Setiap struktur variabel berisi nama sederhana untuk digunakan dalam ekspresi dan **value** struktur yang mengidentifikasi properti mana yang akan ditautkan ke variabel itu. **valueStruktur** berisi informasi berikut:
 - **propertyId**— ID properti dari mana untuk memasukkan nilai. Anda dapat menggunakan nama properti alih-alih ID-nya.

Important

Transformasi dapat memasukkan properti yang integer, ganda, Boolean, atau tipe string. Boolean mengkonversi ke 0 (false) dan 1 (true).

Transformasi harus memasukkan satu atau lebih properti yang bukan atribut dan sejumlah properti atribut. AWS IoT SiteWise menghitung titik data baru yang diubah setiap kali properti input yang bukan atribut menerima titik data baru. Nilai atribut baru tidak meluncurkan pembaruan transformasi. Tingkat permintaan yang sama untuk operasi API data properti aset berlaku untuk hasil komputasi transformasi.

Ekspresi rumus hanya dapat menampilkan nilai ganda atau string. Ekspresi bersarang dapat menampilkan tipe data lain, seperti string, tetapi rumus secara keseluruhan harus mengevaluasi ke angka atau string. Anda dapat menggunakan [fungsi jp](#) untuk mengonversi string ke angka. Nilai Boolean harus 1 (true) atau 0 (false). Untuk informasi selengkapnya, lihat [Nilai tidak terdefinisi, tak terbatas, dan meluap](#).

Example mengubah definisi

Contoh berikut menunjukkan properti transformasi yang mengubah data pengukuran suhu aset dari Celcius ke Fahrenheit. Objek ini adalah contoh dari sebuah [AssetModelProperty](#) yang berisi [Transform](#). Anda dapat menentukan objek ini sebagai bagian dari payload [CreateAssetModel](#) permintaan untuk membuat properti transform. Untuk informasi selengkapnya, lihat [Buat model aset \(AWS CLI\)](#).

```
{
```



```

...
"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 mengubah definisi yang berisi tiga variabel

Contoh berikut menunjukkan properti transform yang mengembalikan pesan peringatan ("Caution") jika lebih dari 10 persen BLT123 bagian tidak memenuhi standar. Jika tidak, ia mengembalikan pesan informasi ("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"
      }
    }
  ]
}
}
...
}

```

Data agregat dari properti dan aset lainnya (metrik)

Metrik adalah ekspresi matematika yang menggunakan fungsi agregasi untuk memproses semua titik data input dan output satu titik data per interval waktu tertentu. Misalnya, metrik dapat menghitung suhu rata-rata per jam dari aliran data suhu.

Metrik dapat memasukkan data dari metrik aset terkait, sehingga Anda dapat menghitung statistik yang memberikan wawasan tentang operasi atau subset operasi Anda. Misalnya, metrik dapat menghitung suhu rata-rata per jam di semua turbin angin di ladang angin. Untuk informasi selengkapnya tentang cara mendefinisikan asosiasi antar aset, lihat [Tentukan hierarki model aset](#).

Metrik juga dapat memasukkan data dari properti lain tanpa menggabungkan data setiap interval waktu. Jika Anda menentukan [atribut](#) dalam rumus, AWS IoT SiteWise gunakan nilai [terbaru](#) untuk atribut tersebut saat menghitung rumus. Jika Anda menentukan metrik dalam rumus, AWS IoT SiteWise gunakan nilai [terakhir](#) untuk interval waktu di mana ia menghitung rumus. Ini berarti Anda dapat menentukan metrik seperti $OEE = Availability * Quality * Performance$, di mana *Availability*, *Quality*, dan *Performance* semua metrik lain pada model aset yang sama.

AWS IoT SiteWise juga secara otomatis menghitung satu set metrik agregasi dasar untuk semua properti aset. Untuk mengurangi biaya komputasi, Anda dapat menggunakan agregat ini alih-alih mendefinisikan metrik khusus untuk perhitungan dasar. Untuk informasi selengkapnya, lihat [Agregat properti aset kueri di AWS IoT SiteWise](#).

Topik

- [Tentukan metrik \(konsol\)](#)
- [Tentukan metrik \(\)AWS CLI](#)

Tentukan metrik (konsol)

Saat menentukan metrik untuk model aset di AWS IoT SiteWise konsol, Anda menentukan parameter berikut:

- Nama - Nama properti.
- Tipe data — Tipe data dari transformasi, yang dapat berupa Double atau String.
- ID Eksternal — (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
- Rumus — Ekspresi metrik. Ekspresi metrik dapat menggunakan [fungsi agregasi](#) untuk memasukkan data dari properti untuk semua aset terkait dalam hierarki. Mulai mengetik atau tekan tombol panah bawah untuk membuka fitur auto complete. Untuk informasi selengkapnya, lihat [Gunakan ekspresi rumus](#).

Important

Metrik hanya dapat berupa properti yang integer, double, Boolean, atau tipe string. Boolean mengkonversi ke 0 (false) dan 1 (true).

Jika Anda menentukan variabel input metrik apa pun dalam ekspresi metrik, input tersebut harus memiliki interval waktu yang sama dengan metrik keluaran.

Ekspresi rumus hanya dapat menampilkan nilai ganda atau string. Ekspresi bersarang dapat menampilkan tipe data lain, seperti string, tetapi rumus secara keseluruhan harus mengevaluasi ke angka atau string. Anda dapat menggunakan [fungsi jp](#) untuk mengonversi string ke angka. Nilai Boolean harus 1 (true) atau 0 (false). Untuk informasi selengkapnya, lihat [Nilai tidak terdefinisi, tak terbatas, dan meluap](#).

- Interval waktu — Interval waktu metrik. AWS IoT SiteWise mendukung interval waktu jendela jatuh berikut, di mana setiap interval dimulai ketika yang sebelumnya berakhir:

- 1 menit — 1 menit, dihitung pada akhir setiap menit (12:00:00 AM, 12:01:00 AM, 12:02:00 AM, dan seterusnya).
- 5 menit — 5 menit, dihitung pada akhir setiap lima menit dimulai pada jam (12:00:00, 12:05:00 AM, 12:10:00 AM, dan seterusnya).
- 15 menit — 15 menit, dihitung pada akhir setiap lima belas menit dimulai pada jam (12:00:00 AM, 12:15:00 AM, 12:30:00 AM, dan seterusnya).
- 1 jam — 1 jam (60 menit), dihitung pada akhir setiap jam di UTC (12:00:00 AM, 01:00:00 AM, 02:00:00 AM, dan seterusnya).
- 1 hari — 1 hari (24 jam), dihitung pada akhir setiap hari di UTC (12:00:00 Senin, 12:00:00 Selasa, dan seterusnya).
- 1 minggu — 1 minggu (7 hari), dihitung pada akhir setiap hari Minggu di UTC (setiap 12:00:00 Senin).
- Interval khusus - Anda dapat memasukkan interval waktu antara satu menit dan seminggu.
- Tanggal offset — (Opsional) Tanggal referensi untuk mengumpulkan data.
- Waktu offset — (Opsional) Waktu referensi untuk mengumpulkan data. Waktu offset harus antara 00:00:00 dan 23:59:59.
- Offset zona waktu - (Opsional) Zona waktu untuk offset. Jika tidak ditentukan, zona waktu offset default adalah Universal Coordinated Time (UTC).

Zona waktu yang didukung

- (UTC+ 00:00) Waktu Terkoordinasi Universal
- (UTC+ 01:00) Waktu Tengah Eropa
- (UTC+ 02:00) Eropa Timur
- (UTC+03:00) Waktu Afrika Timur
- (UTC+ 04:00) Waktu Timur Dekat
- (UTC+ 05:00) Waktu Lahore Pakistan
- (UTC+ 05:30) Waktu Standar India
- (UTC+ 06:00) Waktu Standar Bangladesh
- (UTC+ 07:00) Waktu Standar Vietnam
-
- (UTC+ 09:00) Waktu Standar Jepang
- (UTC+ 09:30) Waktu Tengah Australia

- (UTC+10:00) Waktu Timur Australia
- (UTC+ 11:00) Waktu Standar Solomon
- (UTC+12:00) Waktu Standar Selandia Baru
- (UTC- 11:00) Waktu Kepulauan Midway
- (UTC- 10:00) Waktu Standar Hawaii
- (UTC- 09:00) Waktu Standar Alaska
- (UTC- 08:00) Waktu Standar Pasifik
- (UTC- 07:00) Waktu Standar Phoenix
- (UTC- 06:00) Waktu Standar Pusat
- (UTC- 05:00) Waktu Standar Timur
- (UTC- 04:00) Puerto Riko dan Waktu Kepulauan Virgin AS
- (UTC- 03:00) Waktu Standar Argentina
- (UTC- 02:00) Waktu Georgia Selatan
- (UTC- 01:00) Waktu Afrika Tengah

Example interval waktu khusus dengan offset (konsol)

Contoh berikut menunjukkan kepada Anda cara menentukan interval waktu 12 jam dengan offset pada 20 Februari 2021, pukul 18:30:30 (PST).

Untuk menentukan interval kustom dengan offset

1. Untuk Interval waktu, pilih Interval khusus.
2. Untuk Interval waktu, lakukan salah satu hal berikut:
 - Masuk **12**, lalu pilih jam.
 - Masuk **720**, lalu pilih menit.
 - Masukkan **43200**, lalu pilih detik.

 Important

Interval waktu harus berupa bilangan bulat terlepas dari unit.

3. Untuk tanggal Offset, pilih 2021/02/20.

Tentukan properti data

4. Untuk waktu Offset, masukkan **18:30:30**.
5. Untuk zona waktu Offset, pilih (UTC- 08:00) Waktu Standar Pasifik.

Jika Anda membuat metrik pada 1 Juli 2021, sebelum atau pukul 18:30:30 (PST), Anda mendapatkan hasil agregasi pertama pada 1 Juli 2021, pukul 18.30:30 (PST). Hasil agregasi kedua adalah pada 2 Juli 2021 pukul 06.30:30 WIB, dan seterusnya.

Tentukan metrik ()AWS CLI

Saat menentukan metrik untuk model aset dengan AWS IoT SiteWise API, Anda menentukan parameter berikut:

- **name**— Nama properti.
- **dataType**— Tipe data metrik, yang bisa DOUBLE atau STRING.
- **externalId**— (Opsional) Ini adalah ID yang ditentukan pengguna. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
- **expression**— Ekspresi metrik. Ekspresi metrik dapat menggunakan [fungsi agregasi](#) untuk memasukkan data dari properti untuk semua aset terkait dalam hierarki. Untuk informasi selengkapnya, lihat [Gunakan ekspresi rumus](#).
- **window**— Interval waktu dan offset untuk jendela jatuh metrik, di mana setiap interval dimulai ketika yang sebelumnya berakhir:
 - **interval**— Interval waktu untuk jendela jatuh. Interval waktu harus antara satu menit dan seminggu.
 - **offsets**— Offset untuk jendela yang jatuh.

Untuk informasi selengkapnya, lihat [TumblingWindow](#) di dalam Referensi API AWS IoT SiteWise .

Example interval waktu kustom dengan offset ()AWS CLI

Contoh berikut menunjukkan kepada Anda cara menentukan interval waktu 12 jam dengan offset pada 20 Februari 2021, pukul 06:30:30 (PST).

```
{
  "window": {
    "tumbling": {
      "interval": "12h",
```

```

      "offset": " 2021-07-23T18:30:30-08"
    }
  }
}

```

Jika Anda membuat metrik pada 1 Juli 2021, sebelum atau pukul 18:30:30 (PST), Anda mendapatkan hasil agregasi pertama pada 1 Juli 2021, pukul 18.30:30 (PST). Hasil agregasi kedua adalah pada 2 Juli 2021 pukul 06.30:30 WIB, dan seterusnya.

- **variables**— Daftar variabel yang mendefinisikan properti lain dari aset atau aset anak Anda untuk digunakan dalam ekspresi. Setiap struktur variabel berisi nama sederhana untuk digunakan dalam ekspresi dan **value** struktur yang mengidentifikasi properti mana yang akan ditautkan ke variabel tersebut. **valueStruktur** berisi informasi berikut:
 - **propertyId**— ID properti dari mana untuk menarik nilai. Anda dapat menggunakan nama properti alih-alih ID-nya jika properti didefinisikan dalam model saat ini (bukan didefinisikan dalam model dari hierarki).
 - **hierarchyId**— (Opsional) ID hierarki untuk menanyakan aset turunan untuk properti. Anda dapat menggunakan nama definisi hierarki alih-alih ID-nya. Jika Anda menghilangkan nilai ini, AWS IoT SiteWise temukan properti dalam model saat ini.

Important

Metrik hanya dapat berupa properti yang integer, double, Boolean, atau tipe string. Boolean mengkonversi ke 0 (false) dan 1 (true).

Jika Anda menentukan variabel input metrik apa pun dalam ekspresi metrik, input tersebut harus memiliki interval waktu yang sama dengan metrik keluaran.

Ekspresi rumus hanya dapat menampilkan nilai ganda atau string. Ekspresi bersarang dapat menampilkan tipe data lain, seperti string, tetapi rumus secara keseluruhan harus mengevaluasi ke angka atau string. Anda dapat menggunakan [fungsi jp](#) untuk mengonversi string ke angka. Nilai Boolean harus 1 (true) atau 0 (false). Untuk informasi selengkapnya, lihat [Nilai tidak terdefinisi, tak terbatas, dan meluap](#).

- **unit**— (Opsional) Unit ilmiah untuk properti, seperti mm atau Celcius.

Example Contoh definisi metrik

Contoh berikut menunjukkan properti metrik yang mengumpulkan data pengukuran suhu aset untuk menghitung suhu per jam maksimum di Fahrenheit. Objek ini adalah contoh dari sebuah

[AssetModelProperty](#) yang berisi [Metrik](#). Anda dapat menentukan objek ini sebagai bagian dari payload [CreateAssetModel](#) permintaan untuk membuat properti metrik. Untuk informasi selengkapnya, lihat [Buat model aset \(AWS CLI\)](#).

```
{
  ...
  "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 Contoh definisi metrik yang memasukkan data dari aset terkait

Contoh berikut menunjukkan properti metrik yang mengumpulkan beberapa data daya rata-rata turbin angin untuk menghitung total daya rata-rata untuk ladang angin. Objek ini adalah contoh dari sebuah [AssetModelProperty](#) yang berisi [Metrik](#). Anda dapat menentukan objek ini sebagai bagian dari payload [CreateAssetModel](#) permintaan untuk membuat properti metrik.


```

{
    ...
    "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"
        }
    ],
    ...
}

```

Gunakan ekspresi rumus

Dengan ekspresi rumus, Anda dapat menentukan fungsi matematika untuk mengubah dan menggabungkan data industri mentah Anda untuk mendapatkan wawasan tentang operasi Anda. Ekspresi rumus menggabungkan literal, operator, fungsi, dan variabel untuk memproses data. Untuk informasi selengkapnya tentang cara mendefinisikan properti aset yang menggunakan ekspresi rumus, lihat [Mengubah data \(mengubah\)](#) dan [Data agregat dari properti dan aset lainnya \(metrik\)](#). Transformasi dan metrik adalah properti rumus.

Topik

- [Gunakan variabel dalam ekspresi rumus](#)
- [Gunakan literal dalam ekspresi rumus](#)
- [Gunakan operator dalam ekspresi rumus](#)
- [Gunakan konstanta dalam ekspresi rumus](#)
- [Gunakan fungsi dalam ekspresi rumus](#)
- [Tutorial ekspresi rumus](#)

Gunakan variabel dalam ekspresi rumus

Variabel mewakili properti AWS IoT SiteWise aset dalam ekspresi rumus. Gunakan variabel untuk memasukkan nilai dari properti aset lain dalam ekspresi Anda, sehingga Anda dapat memproses data dari properti konstan ([atribut](#)), aliran data mentah ([pengukuran](#)), dan properti rumus lainnya.

Variabel dapat mewakili properti aset dari model aset yang sama atau dari model aset anak terkait. Hanya rumus metrik yang dapat memasukkan variabel dari model aset anak.

Anda mengidentifikasi variabel dengan nama yang berbeda di konsol dan API.

- AWS IoT SiteWise konsol — Gunakan nama properti aset sebagai variabel dalam ekspresi Anda.
- AWS IoT SiteWise API (AWS CLI, AWS SDKs) — Tentukan variabel dengan [ExpressionVariable](#) struktur, yang memerlukan nama variabel dan referensi ke properti aset. Nama variabel dapat berisi huruf kecil, angka, dan garis bawah. Kemudian, gunakan nama variabel untuk mereferensikan properti aset dalam ekspresi Anda.

Nama variabel peka huruf besar/kecil.

Untuk informasi selengkapnya, lihat [Mendefinisikan transformasi](#) dan [Mendefinisikan](#) metrik.

Gunakan variabel untuk referensi properti

Nilai variabel mendefinisikan properti yang dirujuk. AWS IoT SiteWise menyediakan berbagai cara untuk melakukan ini.

- Berdasarkan ID properti: Anda dapat menentukan ID unik properti (UUID) untuk mengidentifikasinya.
- Dengan nama: Jika properti berada pada model aset yang sama, Anda dapat menentukan namanya di bidang ID properti.

- Dengan jalur: Nilai variabel dapat merujuk ke properti dengan jalurnya. Untuk informasi selengkapnya, lihat [Gunakan jalur untuk mereferensikan properti model komposit khusus](#).

Note

Variabel tidak didukung oleh AWS IoT SiteWise konsol. Mereka digunakan oleh AWS IoT SiteWise API, termasuk AWS Command Line Interface AWS CLI) dan AWS SDKs.

Variabel yang Anda terima dalam respons AWS IoT SiteWise mencakup informasi lengkap tentang nilai, termasuk ID dan jalur.

Namun, ketika Anda meneruskan variabel ke AWS IoT SiteWise (misalnya, dalam panggilan “buat” atau “perbarui”), Anda hanya perlu menentukan salah satunya. Misalnya, jika Anda menentukan jalur, Anda tidak perlu memberikan ID.

Gunakan literal dalam ekspresi rumus

AWS IoT SiteWise mendukung penggunaan literal dalam ekspresi dan rumus. Literal adalah nilai tetap yang mewakili tipe data tertentu. Di AWS IoT SiteWise, Anda dapat menentukan angka dan string literal dalam ekspresi rumus. Literal dapat digunakan dalam berbagai konteks, termasuk transformasi data, kondisi alarm, dan perhitungan visualisasi.

• Angka

Gunakan angka dan notasi ilmiah untuk mendefinisikan bilangan bulat dan ganda. Anda dapat menggunakan [notasi E](#) untuk mengekspresikan angka dengan notasi ilmiah.

Contoh: 1,2.0,.9,-23.1,7.89e3, 3.4E-5

• String

Gunakan karakter ' (kutipan) dan " (kutipan ganda) untuk menentukan string. Jenis kutipan untuk awal dan akhir harus cocok. Untuk menghindari kutipan yang cocok dengan yang Anda gunakan untuk mendeklarasikan string, sertakan karakter kutipan itu dua kali. Ini adalah satu-satunya karakter pelarian dalam AWS IoT SiteWise string.

Contoh: 'active','inactive','{"temp": 52}','{"temp": "high"}'

Gunakan operator dalam ekspresi rumus

Anda dapat menggunakan operator umum berikut dalam ekspresi rumus.

Operator	Deskripsi
+	Jika kedua operan adalah angka, operator ini menambahkan operan kiri dan kanan. Jika salah satu operan adalah string, operator ini menggabungkan operan kiri dan kanan sebagai string. Misalnya, ekspresi <code>1 + 2 + "is three"</code> mengevaluasi. <code>"3 is three"</code> String gabungan dapat memiliki hingga 1024 karakter. Jika string melebihi 1024 karakter, maka AWS IoT SiteWise tidak menampilkan titik data untuk perhitungan itu.
-	Mengurangi operan kanan dari operan kiri. Anda hanya dapat menggunakan operator ini dengan operan numerik.
/	Membagi operan kiri dengan operan kanan. Anda hanya dapat menggunakan operator ini dengan operan numerik.
*	Mengalikan operan kiri dan kanan. Anda hanya dapat menggunakan operator ini dengan operan numerik.
^	Meningkatkan operan kiri ke kekuatan operan kanan (eksponensial). Anda hanya dapat menggunakan operator ini dengan operan numerik.

Operator	Deskripsi
<code>%</code>	Mengembalikan sisanya dari membagi operan kiri dengan operan kanan. Hasilnya memiliki tanda yang sama dengan operan kiri. Perilaku ini berbeda dari operasi modulo. Anda hanya dapat menggunakan operator ini dengan operan numerik.
<code>x < y</code>	Mengembalikan 1 x jika kurang dari y, jika tidak 0.
<code>x > y</code>	Mengembalikan 1 jika x lebih besar dari y, jika tidak 0.
<code>x <= y</code>	Mengembalikan 1 x jika kurang dari atau sama dengan y, jika tidak 0.
<code>x >= y</code>	Mengembalikan 1 jika x lebih besar dari atau sama dengan y, jika tidak 0.
<code>x == y</code>	Mengembalikan 1 x jika sama dengan y, jika tidak 0.
<code>x != y</code>	Mengembalikan 1 x jika tidak sama dengan y, jika tidak 0.
<code>!x</code>	Mengembalikan 1 jika x dievaluasi ke 0 (false), jika tidak 0. x dievaluasi menjadi false jika: • x adalah operan numerik dan dievaluasi. 0 • x dievaluasi ke string kosong. • x dievaluasi ke array kosong. • x dievaluasi untuk None.

Operator	Deskripsi
x and y	Mengembalikan 0 jika x dievaluasi ke 0 (false). Jika tidak, mengembalikan hasil evaluasi dari y xatau y dievaluasi menjadi false jika: • xatau y operan numerik dan dievaluasi menjadi. 0 • xatau y dievaluasi ke string kosong. • xatau y dievaluasi ke array kosong. • xatau y dievaluasi untukNone.
x or y	Mengembalikan 1 jika x dievaluasi ke 1 (true). Jika tidak, mengembalikan hasil evaluasi dari y xatau y dievaluasi menjadi false jika: • xatau y operan numerik dan dievaluasi menjadi. 0 • xatau y dievaluasi ke string kosong. • xatau y dievaluasi ke array kosong. • xatau y dievaluasi untukNone.
not x	Mengembalikan 1 jika x dievaluasi ke 0 (false), jika tidak0. xdievaluasi menjadi false jika: • xadalah operan numerik dan dievaluasi. 0 • xdievaluasi ke string kosong. • xdievaluasi ke array kosong. • xdievaluasi untukNone.

Operator	Deskripsi
<code>[]</code> <code>s[index]</code>	Mengembalikan karakter <code>index</code> pada indeks strings. Ini setara dengan sintaks indeks di Python. Example Contoh • <code>"Hello!"[1]</code> kembaliek. • <code>"Hello!"[-2]</code> kembalio.

Operator	Deskripsi
[] s[start:end:step]	Mengembalikan sepotong strings. Ini setara dengan sintaks slice di Python. Operator ini memiliki argumen berikut: • start— (Opsional) Indeks awal inklusif dari irisan. Default ke 0. • end— (Opsional) Indeks akhir eksklusif dari irisan. Default ke panjang string. • step— (Opsional) Jumlah yang akan bertambah untuk setiap langkah dalam irisan. Misalnya, Anda dapat menentukan 2 untuk mengembalikan irisan dengan setiap karakter lain, atau menentukan -1 untuk membalikkan irisan. Default ke 1. Anda dapat menghilangkan step argumen untuk menggunakan nilai defaultnya. Misalnya, s[1:4:1] setara dengan s[1:4]. Argumen harus bilangan bulat atau tidak ada konstanta. Jika Anda menentukan none, AWS IoT SiteWise menggunakan nilai default untuk argumen itu. Example Contoh • "Hello!"[1:4] kembali"ell". • "Hello!"[:2] kembali"He". • "Hello!"[3:] kembali"lo!". • "Hello!"[:-4] kembali"He". • "Hello!"[:2] kembali"Hlo". • "Hello!"[::-1] kembali"!olleH".

Gunakan konstanta dalam ekspresi rumus

Di AWS IoT SiteWise, Anda dapat menggunakan konstanta dalam ekspresi dan rumus Anda untuk mewakili nilai tetap atau parameter yang telah ditentukan sebelumnya. Konstanta dapat digunakan dalam berbagai konteks, seperti transformasi data, kondisi alarm, atau perhitungan visualisasi. Dengan menggunakan konstanta, Anda dapat menyederhanakan ekspresi Anda dan membuatnya lebih mudah dibaca dan dipelihara.

Anda dapat menggunakan konstanta matematika umum berikut dalam ekspresi Anda. Semua konstanta tidak peka huruf besar/kecil.

Note

Jika Anda mendefinisikan variabel dengan nama yang sama dengan konstanta, variabel mengesampingkan konstanta.

Konstan	Deskripsi
pi	Angka pi (π): 3.141592653589793
e	Angka e: 2.718281828459045
true	Setara dengan angka 1. Pada tahun AWS IoT SiteWise, Boolean mengonversi ke jumlah yang setara.
false	Setara dengan angka 0. Pada tahun AWS IoT SiteWise, Boolean mengonversi ke jumlah yang setara.
none	Setara dengan tidak ada nilai. Anda dapat menggunakan konstanta ini untuk tidak menghasilkan apa pun sebagai hasil dari ekspresi bersyarat .

Gunakan fungsi dalam ekspresi rumus

Anda dapat menggunakan fungsi berikut untuk mengoperasikan data dalam ekspresi rumus Anda.

Transformasi dan metrik mendukung fungsi yang berbeda. Tabel berikut menunjukkan jenis fungsi yang kompatibel dengan setiap jenis properti rumus.

 Note

Anda dapat menyertakan maksimal 10 fungsi dalam ekspresi rumus.

Jenis fungsi	Transformasi	Metrik
Gunakan fungsi umum dalam ekspresi rumus	 Ya	 Ya
Gunakan fungsi perbandingan dalam ekspresi rumus	 Ya	 Ya
Gunakan fungsi kondisional dalam ekspresi rumus	 Ya	 Ya
Gunakan fungsi string dalam ekspresi rumus	 Ya	 Ya
Gunakan fungsi agregasi dalam ekspresi rumus	 Tidak	 Ya

Jenis fungsi	Transformasi	Metrik
Gunakan fungsi temporal dalam ekspresi rumus	 Ya	 Ya
Gunakan fungsi tanggal dan waktu dalam ekspresi rumus	 Ya	 Ya

Sintaks fungsi

Anda dapat menggunakan sintaks berikut untuk membuat fungsi:

Sintaks biasa

Dengan sintaks reguler, nama fungsi diikuti oleh tanda kurung dengan nol atau lebih argumen.

function_name(argument1, argument2, argument3, ...). Misalnya, fungsi dengan sintaks biasa mungkin terlihat seperti `log(x)` dan `contains(s, substring)`.

Sintaks panggilan fungsi seragam (UFCS)

UFCS memungkinkan Anda untuk memanggil fungsi menggunakan sintaks untuk panggilan metode dalam pemrograman berorientasi objek. Dengan UFCS, argumen pertama diikuti oleh dot (`.`), lalu nama fungsi dan argumen yang tersisa (jika ada) di dalam tanda kurung.

argument1.function_name(argument2, argument3, ...). Misalnya, fungsi dengan UFCS mungkin terlihat seperti `x.log()` dan `s.contains(substring)`.

Anda juga dapat menggunakan UFCS untuk merantai fungsi selanjutnya. AWS IoT SiteWise menggunakan hasil evaluasi dari fungsi saat ini sebagai argumen pertama untuk fungsi berikutnya.

Misalnya, Anda dapat menggunakan `message.jp('$status').lower().contains('fail')` sebagai pengganti `contains(lower(jp(message, '$status')), 'fail')`.

Untuk informasi lebih lanjut, kunjungi situs web [D Programming Language](#).

Note

Anda dapat menggunakan UFCS untuk semua AWS IoT SiteWise fungsi. AWS IoT SiteWise fungsi tidak peka huruf besar/kecil. Misalnya, Anda dapat menggunakan `lower(s)` dan secara `Lower(s)` bergantian.

Gunakan fungsi umum dalam ekspresi rumus

Dalam [transformasi](#) dan [metrik](#), Anda dapat menggunakan fungsi berikut untuk menghitung fungsi matematika umum dalam transformasi dan metrik.

Fungsi	Deskripsi
<code>abs(x)</code>	Mengembalikan nilai absolut dari <code>x</code> .
<code>acos(x)</code>	Mengembalikan arccosine dari <code>x</code> .
<code>asin(x)</code>	Mengembalikan arcsine dari <code>x</code> .
<code>atan(x)</code>	Mengembalikan arctangent dari <code>x</code> .
<code>cbrt(x)</code>	Mengembalikan akar kubik <code>x</code> .
<code>ceil(x)</code>	Mengembalikan integer terdekat lebih besar dari <code>x</code> .
<code>cos(x)</code>	Mengembalikan cosinus dari <code>x</code> .
<code>cosh(x)</code>	Mengembalikan kosinus hiperbolik <code>x</code> .
<code>cot(x)</code>	Mengembalikan kotangen dari <code>x</code> .
<code>exp(x)</code>	Kembali <code>e</code> ke kekuatan <code>x</code> .

Fungsi	Deskripsi
<code>expm1(x)</code>	Pengembalian $\exp(x) - 1$. Gunakan fungsi ini untuk menghitung secara lebih akurat $\exp(x) - 1$ untuk nilai kecil x .
<code>floor(x)</code>	Mengembalikan integer terdekat kurang dari x .
<code>log(x)</code>	Mengembalikan $\log_e$ (basis e) dari x .
<code>log10(x)</code>	Mengembalikan $\log_{10}$ (basis 10) dari x .
<code>log1p(x)</code>	Pengembalian $\log(1 + x)$. Gunakan fungsi ini untuk menghitung secara lebih akurat $\log(1 + x)$ untuk nilai kecil x .
<code>log2(x)</code>	Mengembalikan $\log_2$ (basis 2) dari x .
<code>pow(x, y)</code>	Kembali x ke kekuatannya. Ini setara dengan x^y .
<code>signum(x)</code>	Mengembalikan tanda x (-1 untuk input negatif, 0 untuk input nol, +1 untuk input positif).
<code>sin(x)</code>	Mengembalikan sinus dari x .
<code>sinh(x)</code>	Mengembalikan sinus hiperbolik dari x .
<code>sqrt(x)</code>	Mengembalikan akar kuadrat dari x .
<code>tan(x)</code>	Mengembalikan garis singgung x .
<code>tanh(x)</code>	Mengembalikan tangen hiperbolik dari x .

Gunakan fungsi perbandingan dalam ekspresi rumus

Dalam [transformasi](#) dan [metrik](#), Anda dapat menggunakan fungsi perbandingan berikut untuk membandingkan dua nilai dan output 1 (true) atau 0 (false). AWS IoT SiteWise membandingkan string dengan urutan [leksikografi](#).

Fungsi	Deskripsi
<code>gt(x, y)</code>	Mengembalikan 1 jika x lebih besar dariy, jika tidak 0 ($x > y$). Fungsi ini tidak mengembalikan nilai if x dan tipe y yang tidak kompatibel, seperti angka dan string.
<code>gte(x, y)</code>	Mengembalikan 1 jika x lebih besar dari atau sama dengany, jika tidak 0 ($x \geq y$). AWS IoT SiteWise menganggap argumen sama jika mereka berada dalam toleransi relatif1E-9. Ini berperilaku mirip dengan fungsi isclose di Python. Fungsi ini tidak mengembalikan nilai if x dan tipe y yang tidak kompatibel, seperti angka dan string.
<code>eq(x, y)</code>	Mengembalikan 1 x jika sama dengany, jika tidak 0 ($x == y$). AWS IoT SiteWise menganggap argumen sama jika mereka berada dalam toleransi relatif1E-9. Ini berperilaku mirip dengan fungsi isclose di Python. Fungsi ini tidak mengembalikan nilai if x dan tipe y yang tidak kompatibel, seperti angka dan string.
<code>lt(x, y)</code>	Mengembalikan 1 x jika kurang dariy, jika tidak 0 ($x < y$). Fungsi ini tidak mengembalikan nilai if x dan tipe y yang tidak kompatibel, seperti angka dan string.

Fungsi	Deskripsi
<code>lte(x, y)</code>	Mengembalikan 1 x jika kurang dari atau sama dengany, jika tidak 0 ($x \leq y$). AWS IoT SiteWise menganggap argumen sama jika mereka berada dalam toleransi relatif $1E-9$. Ini berperilaku mirip dengan fungsi isclose di Python. Fungsi ini tidak mengembalikan nilai if x dan tipe y yang tidak kompatibel, seperti angka dan string.
<code>isnan(x)</code>	Mengembalikan 1 x jika sama denganNaN, jika tidak0. Fungsi ini tidak mengembalikan nilai jika x adalah string.

Gunakan fungsi kondisional dalam ekspresi rumus

Dalam [transformasi](#) dan [metrik](#), Anda dapat menggunakan fungsi berikut untuk memeriksa kondisi dan mengembalikan hasil yang berbeda, apakah kondisi dievaluasi menjadi benar atau salah.

Fungsi	Deskripsi
<code>if(condition, result_if_true, result_if_false)</code>	Mengevaluasi <code>condition</code> dan mengembalikan <code>result_if_true</code> jika kondisi dievaluasi menjadi benar atau <code>result_if_false</code> jika kondisi dievaluasi. <code>false</code> <code>condition</code> harus berupa angka. Fungsi ini mempertimbangkan 0 dan string kosong sebagai <code>false</code> dan segala sesuatu yang lain (termasukNaN) sebagai<code>true</code>. Boolean mengkonversi ke 0 (<code>false</code>) dan 1 (<code>true</code>).

Fungsi	Deskripsi
	Anda dapat mengembalikan konstanta <code>none</code> dari fungsi ini untuk membuang output untuk kondisi tertentu. Ini berarti Anda dapat memfilter titik data yang tidak memenuhi persyaratan. Untuk informasi selengkapnya, lihat Filter titik data. Example Contoh • <code>if(0, x, y)</code> mengembalikan variabel <code>y</code>. • <code>if(5, x, y)</code> mengembalikan variabel <code>x</code>. • <code>if(gt(temp, 300), x, y)</code> mengembalikan variabel <code>x</code> jika variabel <code>temp</code> lebih besar dari 300. • <code>if(gt(temp, 300), temp, none)</code> mengembalikan variabel <code>temp</code> jika lebih besar dari atau sama dengan 300, atau <code>none</code> (tidak ada nilai) jika <code>temp</code> kurang dari 300. Kami menyarankan Anda menggunakan UFCS untuk fungsi bersyarat bersarang di mana satu atau lebih argumen adalah fungsi bersyarat. Anda dapat menggunakan <code>if(condition, result_if_true)</code> untuk mengevaluasi suatu kondisi dan <code>elif(condition, result_if_true, result_if_false)</code> untuk mengevaluasi kondisi tambahan. Misalnya, Anda dapat menggunakan <code>if(condition1, result1_if_true).elif(condition2, result2_if_true, result2_if_false)</code> sebagai pengganti <code>if(condition1, result1_if_true,</code>

Fungsi	Deskripsi
	<pre>if(condition2, result2_if_true, result2_if_false)) .</pre> Anda juga dapat menghubungkan fungsi kondisional menengah tambahan. Misalnya, Anda dapat menggunakan <code>if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_if_false)</code> alih-alih menyoroti beberapa <code>if</code> pernyataan, seperti <code>if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true, result3_if_false)))</code> .  Important Anda harus menggunakan <code>elif(condition, result_if_true, result_if_false)</code> dengan UFCS.

Gunakan fungsi string dalam ekspresi rumus

Dalam [transformasi](#) dan [metrik](#), Anda dapat menggunakan fungsi berikut untuk beroperasi pada string. Untuk informasi selengkapnya, lihat [Gunakan string dalam rumus](#).

Important

Ekspresi rumus hanya dapat menampilkan nilai ganda atau string. Ekspresi bersarang dapat menampilkan tipe data lain, seperti string, tetapi rumus secara keseluruhan harus mengevaluasi ke angka atau string. Anda dapat menggunakan [fungsi jp](#) untuk mengonversi string ke angka. Nilai Boolean harus 1 (true) atau 0 (false). Untuk informasi selengkapnya, lihat [Nilai tidak terdefinisi, tak terbatas, dan meluap](#).

Fungsi	Deskripsi
<code>len(s)</code>	Mengembalikan panjang strings.
<code>find(s, substring)</code>	Mengembalikan indeks string substring dalam strings.
<code>contains(s, substring)</code>	Mengembalikan 1 jika string s berisi <code>substring</code> , jika tidak 0.
<code>upper(s)</code>	Mengembalikan string s dalam bentuk huruf besar.
<code>lower(s)</code>	Mengembalikan string s dalam bentuk huruf kecil.
<code>jp(s, json_path)</code>	Mengevaluasi string s dengan JsonPath ekspresi <code>json_path</code> dan mengembalikan hasilnya. Gunakan fungsi ini untuk melakukan hal berikut: • Ekstrak nilai, array, atau objek dari struktur JSON serial. • Mengkonversi string ke angka. Misalnya, rumus <code>jp('111', '\$')</code> kembali 111 sebagai angka. Untuk mengekstrak nilai string dari struktur JSON dan mengembalikannya sebagai angka, Anda harus menggunakan beberapa fungsi bersarang <code>jp</code>. <code>jp</code> Fungsi luar mengekstrak string dari struktur JSON, dan <code>jp</code> fungsi dalam mengubah string menjadi angka.

Fungsi	Deskripsi
	String <code>json_path</code> harus berisi string literal. Ini berarti bahwa tidak <code>json_path</code> bisa menjadi ekspresi yang mengevaluasi ke string. Example Contoh • <code>jp('{"status":"active","value":15}', '\$.value')</code> kembali15. • <code>jp('{"measurement":{"reading":25,"confidence":0.95}}', '\$.measurement.reading')</code> kembali25. • <code>jp('[2,8,23]', '\$[2]')</code> kembali23. • <code>jp('{"values":[3,6,7]}', '\$.values[1]')</code> kembali6. • <code>jp('111', '\$')</code>kembali111. • <code>jp(jp('{"measurement":{"reading":25,"confidence":"0.95"}}', '\$.measurement.confidence'), '\$')</code> kembali0.95.
<pre>join(s0, s1, s2, s3, ...)</pre>	Mengembalikan string gabungan dengan pembatas. Fungsi ini menggunakan string input pertama sebagai pembatas dan menggabungkan string input yang tersisa bersama-sama. Ini berperilaku mirip dengan fungsi join (CharSequence delimiter, CharSequence... elements) di Java. Example Contoh • <code>join("-", "aa", "bb", "cc")</code>kembali aa-bb-cc

Fungsi	Deskripsi
<code>format(expression: "format")</code> atau <code>format("format", expression)</code>	Mengembalikan string dalam format yang ditentukan. Fungsi ini mengevaluasi <code>expression</code> nilai, dan kemudian mengembalikan nilai dalam format yang ditentukan. Ini berperilaku mirip dengan fungsi format (String format, Object... args) di Java. Untuk informasi selengkapnya tentang format yang didukung, lihat Konversi di bawah Formatter Kelas di Platform Java, Spesifikasi API Edisi Standar 7. Example Contoh • <code>format(100+1: "d")</code> mengembalikan string,101. • <code>format("The result is %d", 100+1)</code> mengembalikan string,The result is 101.

Fungsi	Deskripsi
<code>f'expression'</code>	Mengembalikan string gabungan. Dengan fungsi diformat ini, Anda dapat menggunakan an ekspresi sederhana untuk menggabungkan dan memformat string. Fungsi-fungsi ini mungkin berisi ekspresi bersarang. Anda dapat menggunakan <code>{}</code> (kurawal kurawal) untuk menginterpolasi ekspresi. Ini berperilaku mirip dengan literal string yang diformat di Python. Example Contoh <code>f'abc{1+2: "f"}d'</code> kembali <code>abc3.000000d</code> . Untuk mengevaluasi ekspresi contoh ini, lakukan hal berikut: <code>format(1+2: "f")</code> mengembalikan nomor floating point, <code>3.000000</code>. <code>join(' ', "abc", 1+2, 'd')</code> mengembalikan string, <code>abc3.000000d</code> . Anda juga dapat menulis ekspresi dengan cara berikut: <code>join(' ', "abc", format(1+2: "f"), 'd')</code> .

Gunakan fungsi agregasi dalam ekspresi rumus

Hanya dalam [metrik](#), Anda dapat menggunakan fungsi berikut yang menggabungkan nilai input setiap interval waktu dan menghitung nilai output tunggal. Fungsi agregasi dapat mengumpulkan data dari aset terkait.

Argumen fungsi agregasi dapat berupa [variabel](#), [literal angka](#), [fungsi temporal](#), ekspresi bersarang, atau fungsi agregasi. Rumus `max(latest(x), latest(y), latest(z))` menggunakan fungsi agregasi sebagai argumen dan mengembalikan nilai saat ini terbesar dari `x`, `y`, dan `z` properti.

Anda dapat menggunakan ekspresi bersarang dalam fungsi agregasi. Bila Anda menggunakan ekspresi bersarang, aturan berikut berlaku:

- Setiap argumen hanya dapat memiliki satu variabel.

Example

Misalnya, $\text{avg}(x \cdot (x-1))$ dan $\text{sum}(x/2) / \text{avg}(y^2)$ didukung.

Misalnya, $\text{min}(x/y)$ tidak didukung.

- Setiap argumen dapat memiliki ekspresi bersarang bertingkat.

Example

Misalnya, $\text{sum}(\text{avg}(x^2)/2)$ didukung.

- Argumen yang berbeda dapat memiliki variabel yang berbeda.

Example

Misalnya, $\text{sum}(x/2, y^2)$ didukung.

Note

- Jika ekspresi Anda berisi pengukuran, AWS IoT SiteWise gunakan nilai terakhir selama interval waktu saat ini untuk pengukuran untuk menghitung agregat.
- Jika ekspresi Anda berisi atribut, AWS IoT SiteWise gunakan nilai terbaru untuk atribut untuk menghitung agregat.

Fungsi	Deskripsi
$\text{avg}(x_0, \dots, x_n)$	Mengembalikan rata-rata nilai variabel yang diberikan selama interval waktu saat ini. Fungsi ini menghasilkan titik data hanya jika variabel yang diberikan memiliki setidaknya satu titik data selama interval waktu saat ini.
$\text{sum}(x_0, \dots, x_n)$	Mengembalikan jumlah nilai variabel yang diberikan selama interval waktu saat ini.

Fungsi	Deskripsi
	Fungsi ini menghasilkan titik data hanya jika variabel yang diberikan memiliki setidaknya satu titik data selama interval waktu saat ini.
$\min(x_0, \dots, x_n)$	Mengembalikan nilai minimum variabel yang diberikan selama interval waktu saat ini. Fungsi ini menghasilkan titik data hanya jika variabel yang diberikan memiliki setidaknya satu titik data selama interval waktu saat ini.
$\max(x_0, \dots, x_n)$	Mengembalikan nilai maksimum variabel yang diberikan selama interval waktu saat ini. Fungsi ini menghasilkan titik data hanya jika variabel yang diberikan memiliki setidaknya satu titik data selama interval waktu saat ini.
$\text{count}(x_0, \dots, x_n)$	Mengembalikan jumlah total titik data untuk variabel yang diberikan selama interval waktu saat ini. Untuk informasi selengkapnya tentang cara menghitung jumlah titik data yang memenuhi suatu kondisi, lihat Hitung titik data yang cocok dengan kondisi. Fungsi ini menghitung titik data untuk setiap interval waktu.
$\text{stdev}(x_0, \dots, x_n)$	Mengembalikan standar deviasi nilai variabel yang diberikan selama interval waktu saat ini. Fungsi ini menghasilkan titik data hanya jika variabel yang diberikan memiliki setidaknya satu titik data selama interval waktu saat ini.

Gunakan fungsi temporal dalam ekspresi rumus

Gunakan fungsi temporal untuk mengembalikan nilai berdasarkan stempel waktu titik data.

Gunakan fungsi temporal dalam metrik

Hanya dalam [metrik](#), Anda dapat menggunakan fungsi berikut yang mengembalikan nilai berdasarkan stempel waktu titik data.

Argumen fungsi temporal harus berupa properti dari model aset lokal atau ekspresi bersarang. Ini berarti Anda tidak dapat menggunakan properti dari model aset anak dalam fungsi temporal.

Anda dapat menggunakan ekspresi bersarang dalam fungsi temporal. Bila Anda menggunakan ekspresi bersarang, aturan berikut berlaku:

- Setiap argumen hanya dapat memiliki satu variabel.

Misalnya, `latest( t*9/5 + 32 )` didukung.

- Argumen tidak bisa berupa fungsi agregasi.

Misalnya, `first( sum(x) )` tidak didukung.

Fungsi	Deskripsi
<code>first(x)</code>	Mengembalikan nilai variabel yang diberikan dengan stempel waktu paling awal selama interval waktu saat ini.
<code>last(x)</code>	Mengembalikan nilai variabel yang diberikan dengan timestamp terbaru selama interval waktu saat ini.
<code>earliest(x)</code>	Mengembalikan nilai terakhir variabel yang diberikan sebelum dimulainya interval waktu saat ini. Fungsi ini menghitung titik data untuk setiap interval waktu, jika properti input memiliki setidaknya satu titik data dalam sejarahnya. Lihat time-range-defintion untuk detail.

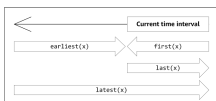
Fungsi	Deskripsi
<code>latest(x)</code>	Mengembalikan nilai terakhir variabel yang diberikan dengan stempel waktu terbaru sebelum akhir interval waktu saat ini. Fungsi ini menghitung titik data untuk setiap interval waktu, jika properti input memiliki setidaknya satu titik data dalam sejarahnya. Lihat time-range-defintion untuk detail.
<code>statetime(x)</code>	Mengembalikan jumlah waktu dalam detik bahwa variabel yang diberikan positif selama interval waktu saat ini. Anda dapat menggunakan an fungsi perbandingan untuk membuat properti transform untuk <code>statetime</code> fungsi yang akan digunakan. Misalnya, jika Anda memiliki <code>Idle</code> properti yang <code>0</code> atau <code>1</code>, Anda dapat menghitung waktu idle per interval waktu dengan ekspresi <code>ini:IdleTime = statetime(Idle)</code> . Untuk informasi selengkapnya, lihat contoh skenario statetime. Fungsi ini tidak mendukung properti metrik sebagai variabel input. Fungsi ini menghitung titik data untuk setiap interval waktu, jika properti input memiliki setidaknya satu titik data dalam sejarahnya.

Fungsi	Deskripsi
<code>TimeWeightedAvg(x, [interpolation])</code>	Mengembalikan rata-rata data masukan yang ditimbang dengan interval waktu antar titik. Lihat Parameter fungsi berbobot waktu untuk detail komputasi dan interval. Argumen opsional <code>interpolation</code> harus berupa konstanta string: • <code>locf</code> Ini adalah defaultnya. Perhitungan menggunakan algoritma komputasi Last Observed Carry Forward untuk interval antar titik data. Dalam pendekatan ini, titik data dihitung sebagai nilai terakhir yang diamati hingga cap waktu titik data input berikutnya. Nilai setelah titik data yang baik diekstrapolasi sebagai nilainya hingga stempel waktu titik data berikutnya. • <code>linear</code>— Perhitungan menggunakan algoritma komputasi interpolasi linier untuk interval antar titik data. Nilai antara dua titik data yang baik diekstrapolasi sebagai interpolasi linier antara nilai titik data tersebut. Nilai antara titik data baik dan buruk atau nilai setelah titik data baik terakhir akan diekstrapolasi sebagai titik data yang baik.

Fungsi	Deskripsi
<code>TimeWeightedStDev(x, [algo])</code>	Mengembalikan standar deviasi data masukan tertimbang dengan interval waktu antar titik. Lihat Parameter fungsi berbobot waktu untuk detail komputasi dan interval. Perhitungan menggunakan algoritma komputasi Last Observed Carry Forward untuk interval antar titik data. Dalam pendekatan ini, titik data dihitung sebagai nilai terakhir yang diamati hingga cap waktu titik data input berikutnya. Berat dihitung sebagai interval waktu dalam hitungan detik antara titik data atau batas jendela. Argumen opsional <code>algo</code> harus berupa konstanta string: • <code>f</code> Ini adalah defaultnya. Ini mengembalikan varians sampel tertimbang yang tidak bias dengan bobot Frekuensi, di mana <code>TimeWeight</code> dihitung dalam hitungan detik. Algoritma ini biasanya diasumsikan di bawah standar deviasi dan dikenal sebagai koreksi standar deviasi Bessel untuk sampel tertimbang. • <code>p</code>— Mengembalikan varians sampel berbobot bias, juga dikenal sebagai Varians populasi. Rumus berikut digunakan untuk perhitungan di mana: • S_p = deviasi standar populasi • S_f = standar deviasi frekuensi • X_i = data masuk

Fungsi	Deskripsi
	• ω_i = berat yang sama dengan interval waktu dalam detik • μ^* = rata-rata tertimbang dari data yang masuk Persamaan untuk standar deviasi populasi: $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$ Persamaan untuk standar deviasi frekuensi: $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

Diagram berikut menunjukkan bagaimana AWS IoT SiteWise menghitung fungsi `temporalfirst`, `last`, dan `earliestlatest`, relatif terhadap interval waktu saat ini.



Note

- Rentang waktu untuk `first(x)`, `last(x)` adalah (mulai jendela saat ini, akhir jendela saat ini].
- Rentang waktu untuk `latest(x)` adalah (awal waktu, akhir jendela saat ini].
- Rentang waktu untuk `earliest(x)` adalah (awal waktu, akhir jendela sebelumnya].

Parameter fungsi berbobot waktu

Fungsi berbobot waktu yang dihitung untuk jendela agregat memperhitungkan hal-hal berikut:

- Titik data di dalam jendela
- Interval waktu antara titik data
- Titik data terakhir sebelum jendela
- Titik data pertama setelah jendela (untuk beberapa algoritma)

Ketentuan:

- Titik data buruk — Setiap titik data dengan kualitas yang tidak baik atau nilai non-angka. Ini tidak dipertimbangkan dalam perhitungan hasil jendela.
- Interval buruk — Interval setelah titik data yang buruk. Interval sebelum titik data pertama yang diketahui juga dianggap sebagai interval yang buruk.
- Titik data yang baik — Setiap titik data dengan kualitas dan nilai numerik yang baik.

 Note

- AWS IoT SiteWise hanya mengkonsumsi data GOOD berkualitas saat menghitung transformasi dan metrik. Ini mengabaikan UNCERTAIN dan titik BAD data.
- Interval sebelum titik data pertama yang diketahui dianggap sebagai interval yang buruk. Untuk informasi selengkapnya, lihat [the section called “Tutorial ekspresi rumus”](#).

Interval setelah titik data terakhir yang diketahui berlanjut tanpa batas, mempengaruhi semua jendela berikut. Ketika titik data baru tiba, fungsi menghitung ulang interval.

Mengikuti aturan di atas, hasil jendela agregat dihitung dan terbatas pada batas jendela. Secara default, fungsi hanya mengirimkan hasil jendela jika seluruh jendela adalah interval yang baik.

Jika interval jendela yang baik lebih kecil dari panjang jendela, fungsi tidak mengirim jendela.

Ketika titik data yang mempengaruhi hasil jendela berubah, fungsi menghitung ulang jendela, bahkan jika titik data berada di luar jendela.

Jika properti input memiliki setidaknya satu titik data dalam histori dan perhitungan telah dimulai, fungsi menghitung fungsi agregat berbobot waktu untuk setiap interval waktu.

Example Contoh skenario statetime

Pertimbangkan contoh di mana Anda memiliki aset dengan properti berikut:

- **Idle**— Pengukuran yang 0 atau 1. Ketika nilainya 1, mesin menganggur.
- **Idle Time**— Metrik yang menggunakan rumus `statetime(Idle)` untuk menghitung jumlah waktu dalam detik di mana mesin dalam keadaan idle, per interval 1 menit.

IdleProperti ini memiliki titik data berikut.

Stempel waktu	14:00:00	14:00:30	14:01:15 SORE	14:02:45 SORE	14:04:00 SORE
Idle	0	1	1	0	0

AWS IoT SiteWise menghitung Idle Time properti setiap menit dari nilai Idle. Setelah perhitungan ini selesai, Idle Time properti memiliki titik data berikut.

Stempel waktu	14:00:00	14:01:00PM	14:02:00PM	14:03:00 SORE	14:04:00 SORE
Idle Time	N/A	30	60	45	0

AWS IoT SiteWise melakukan perhitungan berikut untuk Idle Time pada akhir setiap menit.

- Pukul 14:00 (untuk 13:59 hingga 14:00)
 - Tidak ada data Idle sebelum pukul 14:00, jadi tidak ada titik data yang dihitung.
- Pukul 14:01 (untuk 14:00 PM hingga 14:01 PM)
 - Pada pukul 14:00:00, mesin aktif (Idle adalah 0).
 - Pada 14:00:30, mesin dalam keadaan idle (Idle adalah 1).
 - Idle tidak berubah lagi sebelum akhir interval pada 14:01:00 PM, begitu Idle Time juga 30 detik.
- Pukul 14:02 (untuk 14:01 PM hingga 14:02 PM)
 - Pada pukul 14:01:00, mesin dalam keadaan idle (per titik data terakhir pada pukul 14:00:30).

- Pukul 14:01:15, mesin masih menganggur.
- Idle tidak berubah lagi sebelum akhir interval pada 2:02:00 PM, begitu Idle Time juga 60 detik.
- Pukul 14:03 (untuk 14:02 PM hingga 14:03 PM)
 - Pada pukul 14:02:00, mesin dalam keadaan idle (per titik data terakhir pada 2:01:15 PM).
 - Pada pukul 14:02:45, mesin aktif.
 - Idle tidak berubah lagi sebelum akhir interval pada 2:03:00 PM, begitu Idle Time juga 45 detik.
- Pukul 14:04 (untuk 14:03 PM hingga 14:04 PM)
 - Pada 2:03:00 PM, mesin aktif (per titik data terakhir pada 2:02:45 PM).
 - Idle tidak berubah lagi sebelum akhir interval pada 2:04:00 PM, begitu Idle Time juga 0 detik.

Example Contoh TimeWeightedAvg dan TimeWeightedStDev skenario

Tabel berikut menyediakan contoh input dan output untuk metrik jendela satu menit ini:

`Avg(x)`, `TimeWeightedAvg(x)`, `TimeWeightedAvg(x, "linear")`, `stDev(x)`, `timeWeightedStDev(x)`, `timeWeightedStDev(x, 'p')`

Masukan sampel untuk jendela agregat satu menit:

Note

Semua titik data ini memiliki GOOD kualitas.

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

Output hasil agregat:

Note

Tidak ada - Hasil tidak diproduksi untuk jendela ini.

Waktu	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:00:00	4	Tidak ada	Tidak ada	0	Tidak ada	Tidak ada
3:01:00	2	4	3	0	0	0
3:02:00	14	9	13	6	5.430610041581775	5.385164807134504
3:03:00	11	13	12.875	8.54400374531753	7.724054437220943	7.659416862050705
3:04:00	0	10	2.5	0	10.084389681792215	10
3:05:00	Tidak ada	0	0	Tidak ada	0	0

Gunakan fungsi temporal dalam transformasi

Dalam [transformasi](#) saja, Anda dapat menggunakan `pretrigger()` fungsi untuk mengambil nilai GOOD kualitas untuk variabel sebelum pembaruan properti yang memulai perhitungan transformasi saat ini.

Pertimbangkan contoh di mana produsen menggunakan AWS IoT SiteWise untuk memantau status mesin. Pabrik menggunakan pengukuran dan transformasi berikut untuk mewakili proses:

- Pengukuran, `current_state`, yang bisa 0 atau 1.
 - Jika mesin dalam keadaan pembersihan, `current_state` sama dengan 1.
 - Jika mesin dalam keadaan manufaktur, `current_state` sama dengan 0.
- Sebuah transformasi, `cleaning_state_duration`, itu sama.
`if(pretrigger(current_state) == 1, timestamp(current_state) - timestamp(pretrigger(current_state)), none)` Transformasi ini mengembalikan berapa lama mesin berada dalam keadaan pembersihan dalam hitungan detik, dalam format epoch Unix. Untuk informasi selengkapnya, lihat [Gunakan fungsi kondisional dalam ekspresi rumus](#) dan fungsi [timestamp\(\)](#).

Jika mesin tetap dalam kondisi pembersihan lebih lama dari yang diharapkan, pabrik mungkin menyelidiki mesin.

Anda juga dapat menggunakan `pretrigger()` fungsi dalam transformasi multivariat. Misalnya, Anda memiliki dua pengukuran bernama `x` dan `y`, dan transformasi, `z`, yang sama. `x + y + pretrigger(y)` Tabel berikut menunjukkan nilai untuk `x`, `y`, dan `z` dari 9:00 AM sampai 9:15 AM.

Note

- Contoh ini mengasumsikan bahwa nilai untuk pengukuran tiba secara kronologis. Misalnya, nilai `x` untuk 09:00 AM tiba sebelum nilai `x` untuk 09:05 AM.
- Jika titik data untuk 9:05 tiba sebelum titik data untuk 9:00 AM, `z` tidak dihitung pada 9:05 AM.
- Jika nilai `x` untuk 9:05 tiba sebelum nilai `x` untuk 09:00 AM dan nilai `y` tiba secara kronologis, `z` sama $22 = 20 + 1 + 1$ dengan pukul 9:05.

	09:00AM	09:05AM	09:10AM	09:15AM
x	10	20		30
y	1	2	3	

	09:00AM	09:05AM	09:10AM	09:15AM
$z = x + y$ $+ \text{pretrigge}$ $r(y)$	ytidak menerima titik data apa pun sebelum pukul 09:00. Oleh karena itu, z tidak dihitung pada pukul 09:00.	$23 = 20 + 2 + 1$ pretrigge $r(y)$ sama dengan 1.	$25 = 20 + 3 + 2$ xtidak menerima titik data baru. pretrigge $r(y)$ sama dengan 2.	$36 = 30 + 3 + 3$ ytidak menerima titik data baru. Oleh karena itu, pretrigge $r(y)$ sama dengan 3 pada 09:15 AM.

Gunakan fungsi tanggal dan waktu dalam ekspresi rumus

Dalam [transformasi](#) dan [metrik](#), Anda dapat menggunakan fungsi tanggal dan waktu dengan cara berikut:

- Ambil stempel waktu saat ini dari titik data di UTC atau di zona waktu lokal.
- Membangun stempel waktu dengan argumen, seperti, `year`, `month` dan `day_of_month`
- Ekstrak periode waktu seperti satu tahun atau bulan dengan `unix_time` argumen.

Fungsi	Deskripsi
<code>now()</code>	Mengembalikan tanggal dan waktu saat ini, dalam detik, dalam format epoch Unix.
<code>timestamp()</code>	• Dalam transformasi, fungsi mengembalikan stempel waktu, dalam hitungan detik, dari pesan input dalam format epoch Unix. Hanya dalam transformasi, Anda dapat melakukan salah satu dari yang berikut: • Berikan variabel sebagai argumen untuk fungsi. <code>timestamp(<i>variable-name</i>)</code> Fungsi mengembalikan stempel waktu, dalam hitungan detik, dari nilai

Fungsi	Deskripsi
	GOOD kualitas terbaru untuk variabel yang ditentukan dalam format epoch Unix. Misalnya, jika aset Anda memiliki properti transform bernama <code>Temperature_F</code> yang menggunakan $9/5 * \text{Temperature\_C}$ rumus untuk mengonversi setiap titik data suhu dari Celcius ke Fahrenheit, Anda dapat menggunakan <code>timestamp (Temperature_F)</code> fungsi tersebut untuk mendapatkan stempel waktu nilai kualitas terbaru untuk properti tersebut. <code>GOOD Temperature_F</code> - Gunakan <code>pretrigger()</code> fungsi sebagai argumen untuk fungsi. <code>timestamp (pretrigger(<i>variable-name</i>))</code> Fungsi mengembalikan stempel waktu, dalam hitungan detik, dari nilai GOOD kualitas untuk variabel tertentu sebelum pembaruan properti yang memulai perhitungan transformasi saat ini dalam format epoch Unix. Untuk informasi selengkapnya, lihat Gunakan fungsi temporal dalam transformasi. - Dalam metrik, fungsi mengembalikan stempel waktu yang diambil di akhir jendela saat ini, dalam hitungan detik, dalam format epoch Unix.

Fungsi	Deskripsi
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	Mengembalikan waktu input dalam hitungan detik, dalam format epoch Unix. Persyaratan berikut berlaku untuk menggunakan fungsi ini: • Argumen zona waktu harus berupa string yang diketip (' UTC '). Jika tidak ditentukan, zona waktu default adalah UTC. Argumen zona waktu dapat menjadi argumen pertama atau terakhir. • Argumen tahun, bulan, hari bulan, jam, menit, dan kedua harus berurutan. • Argumen tahun, bulan, dan tanggal diperlukan. Batasan berikut berlaku untuk menggunakan fungsi ini: • <code>year</code>- Nilai yang valid adalah antara 1970 dan 2250. • <code>month</code>- Nilai yang valid adalah antara 1 dan 12. • <code>day-of-month</code> - Nilai yang valid adalah antara 1 - 31. • <code>hour</code>- Nilai yang valid adalah antara 0 dan 23. • <code>minute</code>- Nilai yang valid adalah antara 0 dan 59. • <code>second</code>- Nilai yang valid adalah antara 0 dan 60. Ini bisa menjadi nomor floating point. Contoh:

Fungsi	Deskripsi
	• <code>mktime(2020, 2, 29)</code> • <code>mktime('UTC+3', 2021, 12, 31, 22)</code> • <code>mktime(2022, 10, 13, 2, 55, 13.68, 'PST')</code>

Fungsi	Deskripsi
<code>localtime(unix_time, time_zone)</code>	Mengembalikan tahun, hari dalam sebulan, hari dalam seminggu, hari dalam setahun, jam, menit, atau yang kedua di zona waktu yang ditentukan dari waktu Unix. Persyaratan berikut berlaku untuk menggunakan fungsi ini: - Argumen zona waktu harus berupa string yang dikutip (' UTC '). Jika tidak ditentukan, zona waktu default adalah UTC. - Argumen waktu Unix adalah waktu dalam detik, dalam format epoch Unix. Kisaran valid adalah antara 1-31556889864403199. Ini bisa menjadi nomor floating point. Contoh respon: 2007-12-03T10:15:30+01:00[Europe/Paris] <code>localtime(unix_time, time_zone)</code> bukan fungsi mandiri. <code>sec()</code> Fungsi <code>year()</code> <code>mon()</code> <code>mday</code> <code>wday()</code> <code>yday()</code> <code>hour()</code> <code>minute()</code>,, dan diambil <code>localtime(unix_time, time_zone)</code> sebagai argumen. Contoh: <code>year(localtime('GMT', 1605898608.8113723))</code> <code>now().localtime().year()</code> <code>timestamp().localtime('PST').year()</code>

Fungsi	Deskripsi
	<code>localtime(1605289736, 'Europe/London').year()</code>
<code>year(localtime(unix_time, time_zone))</code>	Mengembalikan tahun dari <code>localtime(unix_time, time_zone)</code> .
<code>mon(localtime(unix_time, time_zone))</code>	Mengembalikan bulan dari <code>localtime(unix_time, time_zone)</code> .
<code>mday(localtime(unix_time, time_zone))</code>	Mengembalikan hari dalam sebulan dari <code>localtime(unix_time, time_zone)</code> .
<code>wday(localtime(unix_time, time_zone))</code>	Mengembalikan hari dalam seminggu dari <code>localtime(unix_time, time_zone)</code> .
<code>yday(localtime(unix_time, time_zone))</code>	Mengembalikan hari dalam setahun dari <code>localtime(unix_time, time_zone)</code> .
<code>hour(localtime(unix_time, time_zone))</code>	Mengembalikan jam dari <code>localtime(unix_time, time_zone)</code> .
<code>minute(localtime(unix_time, time_zone))</code>	Mengembalikan menit dari <code>localtime(unix_time, time_zone)</code> .
<code>sec(localtime(unix_time, time_zone))</code>	Mengembalikan yang kedua dari <code>localtime(unix_time, time_zone)</code> .

Format zona waktu yang didukung

Anda dapat menentukan argumen zona waktu dengan cara berikut:

- Offset zona waktu - Tentukan 'Z' untuk UTC atau offset ('+2' atau. '-5')
- Offset IDs - Gabungkan singkatan zona waktu dan offset. Misalnya, 'GMT+2' dan 'UTC-01:00'. Singkatan zona waktu harus berisi hanya tiga huruf.

- Berbasis wilayah IDs - Misalnya, 'Etc/GMT+12' dan 'Pacific/Pago_Pago'.

Singkatan zona waktu yang didukung

Fungsi tanggal dan waktu mendukung singkatan zona waktu tiga huruf berikut:

- EST - - 05:00
- HST - - 10:00
- MST - - 07:00
- ACT - Australia/Darwin
- AET - Australia/Sydney
- AGT - America/Argentina/Buenos Aires
- ART - Afrika/Kairo
- AST - Amerika/Anchorage
- TARUHAN - Amerika/Sao_Paulo
- BST - Asia/Dhaka
- CAT - Afrika/Harare
- CET - Eropa/Paris
- CNT - Amerika/St_Johns
- CST - Amerika/Chicago
- CTT - Asia/Shanghai
- MAKAN - Afrika/Addis_Ababa
- IET - America/Indiana/Indianapolis
- IST - Asia/Kolkata
- JST - Asia/Tokyo
- MIT - Pasifik/Apia
- NET - Asia/Yerevan
- NST - Pasifik/Auckland
- PLT - Asia/Karachi
- PRT - Amerika/Puerto_Rico

- PST - Amerika/Los_Angeles
- SST - Pasifik/Guadalkanal
- VST - Asia/Ho_Chi_Minh

Berbasis wilayah yang didukung IDs

Fungsi tanggal dan waktu mendukung berbasis Wilayah berikut IDs, yang diatur berdasarkan hubungannya dengan UTC+ 00:00:

- DII/GMT+12 (UTC- 12:00)
- Pasifik/Pago_Pago (UTC- 11:00)
- Pasifik/Samoa (UTC- 11:00)
- Pasifik/Niue (UTC- 11:00)
- AS/Samoa (UTC- 11:00)
- DII/GMT+11 (UTC- 11:00)
- Pasifik/Midway (UTC- 11:00)
- Pasifik/Honolulu (UTC- 10:00)
- Pasifik/Rarotonga (UTC- 10:00)
- Pasifik/Tahiti (UTC- 10:00)
- Pasifik/Johnston (UTC- 10:00)
- AS/Hawaii (UTC- 10:00)
- SystemV/ HST1 0 (UTC- 10:00)
- DII/GMT+10 (UTC- 10:00)
- Pasifik/Marquesas (UTC- 09:30)
- DII/GMT+9 (UTC- 09:00)
- Pasifik/Gambier (UTC- 09:00)
- Amerika/Atka (UTC- 09:00)
- SystemV/ YST9 (UTC- 09:00)
- Amerika/Adak (UTC- 09:00)
- AS/Aleutian (UTC- 09:00)

- DII/GMT+8 (UTC- 08:00)
- AS/Alaska (UTC- 08:00)
- Amerika/Juneau (UTC- 08:00)
- Amerika/Metlakatla (UTC- 08:00)
- Amerika/Yakutat (UTC- 08:00)
- Pasifik/Pitcairn (UTC- 08:00)
- Amerika/Sitka (UTC- 08:00)
- Amerika/Anchorage (UTC- 08:00)
- SystemV/ PST8 (UTC- 08:00)
- Amerika/Nama (UTC- 08:00)
- YST9SystemV/YDT (UTC- 08:00)
- Kanada/Yukon (UTC- 07:00)
- AS/Pasifik-Baru (UTC- 07:00)
- DII/GMT+7 (UTC- 07:00)
- Amerika Serikat/Arizona (UTC- 07:00)
- Amerika/Dawson_Creek (UTC- 07:00)
- Kanada/Pasifik (UTC- 07:00)
- PST8PDT (UTC- 07:00)
- SystemV/ MST7 (UTC- 07:00)
- Amerika/Dawson (UTC- 07:00)
- Meksiko/ BajaNorte (UTC- 07:00)
- Amerika/Tijuana (UTC- 07:00)
- Amerika/Creston (UTC- 07:00)
- Amerika/Hermosillo (UTC- 07:00)
- Amerika/Santa_Isabel (UTC- 07:00)
- Amerika/Vancouver (UTC- 07:00)
- Amerika/Ensenada (UTC- 07:00)
- Amerika/Phoenix (UTC- 07:00)

- Amerika/Whitehorse (UTC- 07:00)
- Amerika/Fort_Nelson (UTC- 07:00)
- PST8SystemV/PDT (UTC- 07:00)
- Amerika/Los_Angeles (UTC- 07:00)
- AS/Pasifik (UTC- 07:00)
- Amerika/El_Salvador (UTC- 06:00)
- Amerika/Guatemala (UTC- 06:00)
- Amerika/Belize (UTC- 06:00)
- Amerika/Managua (UTC- 06:00)
- Amerika/Tegucigalpa (UTC- 06:00)
- DII/GMT+6 (UTC- 06:00)
- Pasifik/Paskah (UTC- 06:00)
- Meksiko/ BajaSur (UTC- 06:00)
- Amerika/Regina (UTC- 06:00)
- Amerika/Denver (UTC- 06:00)
- Pasifik/Galapagos (UTC- 06:00)
- Amerika/Yellowknife (UTC- 06:00)
- Amerika/Swift_Current (UTC- 06:00)
- Amerika/Inuvik (UTC- 06:00)
- Amerika/Mazatlan (UTC- 06:00)
- Amerika/Boise (UTC- 06:00)
- Amerika/Costa_Rica (UTC- 06:00)
- MST7MDT (UTC- 06:00)
- SystemV/ CST6 (UTC- 06:00)
- Amerika/Chihuahua (UTC- 06:00)
- Amerika/Ojinaga (UTC- 06:00)
- Chili/ EasterIsland (UTC- 06:00)
- AS/Gunung (UTC- 06:00)
- Amerika/Edmonton (UTC- 06:00)

- Kanada/Gunung (UTC- 06:00)
- Amerika/Cambridge_Bay (UTC- 06:00)
- Navajo (UTC- 06:00)
- MST7SistemV/MDT (UTC- 06:00)
- Kanada/Saskatchewan (UTC- 06:00)
- Amerika/Shirock (UTC- 06:00)
- Amerika/Panama (UTC- 05:00)
- Amerika/Chicago (UTC- 05:00)
- Amerika/Eirunepe (UTC- 05:00)
- DII/GMT+5 (UTC- 05:00)
- Meksiko/Umum (UTC- 05:00)
- Amerika/Porto_Acre (UTC- 05:00)
- Amerika/Guayaquil (UTC- 05:00)
- Amerika/Rankin_Inlet (UTC- 05:00)
- AS/Tengah (UTC- 05:00)
- Amerika/Rainy_River (UTC- 05:00)
- America/Indiana/Knox(UTC- 05:00)
- America/North_Dakota/Beulah(UTC- 05:00)
- Amerika/Monterrey (UTC- 05:00)
- Amerika/Jamaika (UTC- 05:00)
- Amerika/Atikokan (UTC- 05:00)
- Amerika/Coral_Harbour (UTC- 05:00)
- America/North_Dakota/Center(UTC- 05:00)
- Amerika/Cayman (UTC- 05:00)
- America/Indiana/Tell_Kota (UTC- 05:00)
- Amerika/Meksiko_Kota (UTC- 05:00)
- Amerika/Matamoros (UTC- 05:00)
- CST6CDT (UTC- 05:00)
- Amerika/Knox_in (UTC- 05:00)

- Amerika/Bogota (UTC- 05:00)
- Amerika/Menominey (UTC- 05:00)
- Amerika/Tegas (UTC- 05:00)
- SystemV/ EST5 (UTC- 05:00)
- Kanada/Pusat (UTC- 05:00)
- Brasil/Acre (UTC- 05:00)
- Amerika/Cancun (UTC- 05:00)
- Amerika/Lima (UTC- 05:00)
- Amerika/Bahia_Banderas (UTC- 05:00)
- Amerika Serikat/Indiana (UTC- 05:00)
- Amerika/Rio_Branco (UTC- 05:00)
- CST6SystemV/CDT (UTC- 05:00)
- Jamaika (UTC- 05:00)
- Amerika/Merida (UTC- 05:00)
- America/North_Dakota/New_Salem (UTC- 05:00)
- Amerika/Winnipeg (UTC- 05:00)
- Amerika/Cuiaba (UTC- 04:00)
- Amerika/Marigot (UTC- 04:00)
- America/Indiana/Petersburg(UTC- 04:00)
- Chili/Kontinental (UTC- 04:00)
- Amerika/Grand_Turk (UTC- 04:00)
- Kuba (UTC- 04:00)
- DII/GMT+4 (UTC- 04:00)
- Amerika/Manaus (UTC- 04:00)
- Amerika/Fort_Wayne (UTC- 04:00)
- Amerika/St_Thomas (UTC- 04:00)
- Amerika/Anguilla (UTC- 04:00)
- Amerika/Havana (UTC- 04:00)
- AS/Michigan (UTC- 04:00)

- Amerika/Barbados (UTC- 04:00)
- Amerika/Louisville (UTC- 04:00)
- Amerika/Curacao (UTC- 04:00)
- Amerika/Guyana (UTC- 04:00)
- Amerika/Martinik (UTC- 04:00)
- Amerika/Puerto_Rico (UTC- 04:00)
- Amerika/PORT_OF_Spanyol (UTC- 04:00)
- SystemV/ AST4 (UTC- 04:00)
- America/Indiana/Vevay(UTC- 04:00)
- America/Indiana/Vincennes(UTC- 04:00)
- Amerika/Kralendijk (UTC- 04:00)
- Amerika/Antigua (UTC- 04:00)
- Amerika/Indianapolis (UTC- 04:00)
- Amerika/Iqaluit (UTC- 04:00)
- Amerika/St_Vincent (UTC- 04:00)
- America/Kentucky/Louisville(UTC- 04:00)
- Amerika/Dominika (UTC- 04:00)
- Amerika/Asuncion (UTC- 04:00)
- EST5EDT (UTC- 04:00)
- Amerika/Nassau (UTC- 04:00)
- America/Kentucky/Monticello(UTC- 04:00)
- Brasil/Barat (UTC- 04:00)
- Amerika/Aruba (UTC- 04:00)
- America/Indiana/Indianapolis(UTC- 04:00)
- Amerika/Santiago (UTC- 04:00)
- Amerika/La_Paz (UTC- 04:00)
- Amerika/Thunder_Bay (UTC- 04:00)
- America/Indiana/Marengo(UTC- 04:00)
- Amerika/Blanc-Sablon (UTC- 04:00)

- Amerika/Santo_Domingo (UTC- 04:00)
- AS/Timur (UTC- 04:00)
- Kanada/Timur (UTC- 04:00)
- Amerika/ Port-au-Prince (UTC- 04:00)
- Amerika/St_Barthelemy (UTC- 04:00)
- Amerika/Nipigon (UTC- 04:00)
- Amerika Serikat/Indiana Timur (UTC- 04:00)
- Amerika/St_Lucia (UTC- 04:00)
- Amerika/Montserrat (UTC- 04:00)
- Amerika/Lower_Princes (UTC- 04:00)
- Amerika/Detroit (UTC- 04:00)
- Amerika/Tortola (UTC- 04:00)
- Amerika/Porto_Velho (UTC- 04:00)
- Amerika/Campo_Grande (UTC- 04:00)
- Amerika/Perawan (UTC- 04:00)
- Amerika/Pangnirtung (UTC- 04:00)
- Amerika/Montreal (UTC- 04:00)
- America/Indiana/Winamac(UTC- 04:00)
- Amerika/Boa_Vista (UTC- 04:00)
- Amerika/Grenada (UTC- 04:00)
- Amerika/New_York (UTC- 04:00)
- Amerika/St_Kitts (UTC- 04:00)
- Amerika/Caracas (UTC- 04:00)
- Amerika/Guadeloupe (UTC- 04:00)
- Amerika/Toronto (UTC- 04:00)
- EST5SystemV/EDT (UTC- 04:00)
- America/Argentina/Catamarca(UTC- 03:00)
- Kanada/Atlantik (UTC- 03:00)
- America/Argentina/Cordoba(UTC- 03:00)

- Amerika/Araguaina (UTC- 03:00)
- America/Argentina/Salta(UTC- 03:00)
- DII/GMT+3 (UTC- 03:00)
- Amerika/Montevideo (UTC- 03:00)
- Brasil/Timur (UTC- 03:00)
- America/Argentina/Mendoza(UTC- 03:00)
- America/Argentina/Rio_Gallegos (UTC- 03:00)
- Amerika/Catamarca (UTC- 03:00)
- Amerika/Cordoba (UTC- 03:00)
- Amerika/Sao_Paulo (UTC- 03:00)
- America/Argentina/Jujuy(UTC- 03:00)
- Amerika/Cayenne (UTC- 03:00)
- Amerika/Recife (UTC- 03:00)
- Amerika/Buenos_Aires (UTC- 03:00)
- Amerika/Paramaribo (UTC- 03:00)
- Amerika/Moncton (UTC- 03:00)
- Amerika/Mendoza (UTC- 03:00)
- Amerika/Santarem (UTC- 03:00)
- Atlantik/Bermuda (UTC- 03:00)
- Amerika/Maceio (UTC- 03:00)
- Atlantik/Stanley (UTC- 03:00)
- Amerika/Halifax (UTC- 03:00)
- Antartika/Rothera (UTC- 03:00)
- America/Argentina/San_Luis (UTC- 03:00)
- America/Argentina/Ushuaia(UTC- 03:00)
- Antartika/Palmer (UTC- 03:00)
- Amerika/Punta_Arenas (UTC- 03:00)
- Amerika/Glace_Bay (UTC- 03:00)
- Amerika/Fortaleza (UTC- 03:00)

- Amerika/Thule (UTC- 03:00)
- America/Argentina/La_Rioja (UTC- 03:00)
- Amerika/Belem (UTC- 03:00)
- Amerika/Jujuy (UTC- 03:00)
- Amerika/Bahia (UTC- 03:00)
- Amerika/Goose_Bay (UTC- 03:00)
- America/Argentina/San_Juan (UTC- 03:00)
- America/Argentina/ComodRivadavia(UTC- 03:00)
- America/Argentina/Tucuman(UTC- 03:00)
- Amerika/Rosario (UTC- 03:00)
- AST4SystemV/ADT (UTC- 03:00)
- America/Argentina/Buenos_Aires (UTC- 03:00)
- Amerika/St_Johns (UTC- 02:30)
- Kanada/Newfoundland (UTC- 02:30)
- Amerika/Miquelon (UTC- 02:00)
- DII/GMT+2 (UTC- 02:00)
- Amerika/Godthab (UTC- 02:00)
- Amerika/Noronha (UTC- 02:00)
- Brasil/ DeNoronha (UTC- 02:00)
- Atlantik/South_Georgia (UTC- 02:00)
- DII/GMT+1 (UTC- 01:00)
- Atlantik/Cape_Verde (UTC- 01:00)
- Pasifik/Kiritimati (UTC+ 14:00)
- DII/GMT-14 (UTC+ 14:00)
- Pasifik/Fakaofu (UTC+ 13:00)
- Pasifik/Enderbury (UTC+ 13:00)
- Pasifik/Apia (UTC+ 13:00)
- Pasifik/Tongatapu (UTC+ 13:00)
- DII/GMT-13 (UTC+13:00)

- NZ-OBROLAN (UTC+ 12:45)
- Pasifik/Chatham (UTC+ 12:45)
- Pasifik/Kwajalein (UTC+ 12:00)
- Antartika/ (UTC+ 12:00) McMurdo
- Pasifik/Wallis (UTC+ 12:00)
- Pasifik/Fiji (UTC+ 12:00)
- Pasifik/Funafuti (UTC+ 12:00)
- Pasifik/Nauru (UTC+ 12:00)
- Kwajalein (UTC+ 12:00)
- NZ (UTC+ 12:00)
- Pasifik/Bangun (UTC+ 12:00)
- Antartika/South_Pole (UTC+ 12:00)
- Pasifik/Tarawa (UTC+ 12:00)
- Pasifik/Auckland (UTC+ 12:00)
- Asia/Kamchatka (UTC+ 12:00)
- DII/GMT-12 (UTC+12:00)
- Asia/Anadyr (UTC+ 12:00)
- Pasifik/Majuro (UTC+ 12:00)
- Pasifik/Ponape (UTC+ 11:00)
- Pasifik/Bougainville (UTC+ 11:00)
- Antartika/Macquarie (UTC+ 11:00)
- Pasifik/Pohnpei (UTC+ 11:00)
- Pasifik/Efate (UTC+ 11:00)
- Pasifik/Norfolk (UTC+ 11:00)
- Asia/Magadan (UTC+ 11:00)
- Pasifik/Kosrae (UTC+ 11:00)
- Asia/Sakhalin (UTC+ 11:00)
- Pasifik/Noumea (UTC+ 11:00)
- DII/GMT-11 (UTC+ 11:00)

- Asia/Srednekolymysk (UTC+ 11:00)
- Pasifik/Guadalkanal (UTC+ 11:00)
- Australia/Lord_Howe (UTC+ 10:30)
- Australia/LHI (UTC+ 10:30)
- Australia/Hobart (UTC+ 10:00)
- Pasifik/Yap (UTC+ 10:00)
- Australia/Tasmania (UTC+ 10:00)
- Pasifik/Port_Moresby (UTC+ 10:00)
- Australia/ACT (UTC+10:00)
- Australia/Victoria (UTC+ 10:00)
- Pasifik/Chuuk (UTC+ 10:00)
- Australia/Queensland (UTC+ 10:00)
- Australia/Canberra (UTC+ 10:00)
- Australia/Currie (UTC+ 10:00)
- Pasifik/Guam (UTC+ 10:00)
- Pasifik/Truk (UTC+ 10:00)
- Australia/NSW (UTC+ 10:00)
- Asia/Vladivostok (UTC+ 10:00)
- Pasifik/Saipan (UTC+ 10:00)
- Antartika/Dumont (UTC+ 10:00DURville)
- Australia/Sydney (UTC+ 10:00)
- Australia/Brisbane (UTC+ 10:00)
- DII/GMT-10 (UTC+10:00)
- Asia/Ust-Nera (UTC+ 10:00)
- Australia/Melbourne (UTC+ 10:00)
- Australia/Lindeman (UTC+ 10:00)
- Australia/Utara (UTC+ 09:30)
- Australia/Yancowinna (UTC+ 09:30)
- Australia/Adelaide (UTC+ 09:30)

- Australia/Broken_Hill (UTC+ 09:30)
- Australia/Selatan (UTC+ 09:30)
- Australia/Darwin (UTC+ 09:30)
- DII/GMT-9 (UTC+ 09:00)
- Pasifik/Palau (UTC+ 09:00)
- Asia/Chita (UTC+ 09:00)
- Asia/Dili (UTC+ 09:00)
- Asia/Jayapura (UTC+ 09:00)
- Asia/Yakutsk (UTC+ 09:00)
- Asia/Pyongyang (UTC+ 09:00)
- ROK (UTC+ 09:00)
- Asia/Seoul (UTC+ 09:00)
- Asia/Khandyga (UTC+ 09:00)
- Jepang (UTC+ 09:00)
- Asia/Tokyo (UTC+ 09:00)
- Australia/Eucla (UTC+ 08:45)
- Asia/Kuching (UTC+ 08:00)
- Asia/Chungking (UTC+ 08:00)
- DII/GMT-8 (UTC+08:00)
- Australia/Perth (UTC+ 08:00)
- Asia/Makau (UTC+ 08:00)
- Asia/Makau (UTC+ 08:00)
- Asia/Choibalsan (UTC+ 08:00)
- Asia/Shanghai (UTC+ 08:00)
- Antartika/Casey (UTC+ 08:00)
- Asia/Ulan_Bator (UTC+ 08:00)
- Asia/Chongqing (UTC+ 08:00)
- Asia/Ulaanbaatar (UTC+ 08:00)
- Asia/Taipei (UTC+ 08:00)

- Asia/Manila (UTC+ 08:00)
- PRC (UTC+ 08:00)
- Asia/Ujung_Pandang (UTC+ 08:00)
- Asia/Harbin (UTC+ 08:00)
- Singapura (UTC+ 08:00)
- Asia/Brunei (UTC+ 08:00)
- Australia/Barat (UTC+ 08:00)
- Asia/Hong Kong (UTC+ 08:00)
- Asia/Makassar (UTC+ 08:00)
- Hong Kong (UTC+ 08:00)
- Asia/Kuala_Lumpur (UTC+ 08:00)
- Asia/Irkutsk (UTC+ 08:00)
- Asia/Singapura (UTC+ 08:00)
- Asia/Pontianak (UTC+ 07:00)
- DII/GMT-7 (UTC+ 07:00)
- Asia/Phnom_Penh (UTC+ 07:00)
- Asia/Novosibirsk (UTC+ 07:00)
- Antartika/Davis (UTC+ 07:00)
- Asia/Tomsk (UTC+ 07:00)
- Asia/Jakarta (UTC+ 07:00)
- Asia/Barnaul (UTC+ 07:00)
- India/Natal (UTC+ 07:00)
- Asia/Ho_Chi_Minh (UTC+ 07:00)
- Asia/Hovd (UTC+ 07:00)
- Asia/Bangkok (UTC+ 07:00)
- Asia/Vientiane (UTC+ 07:00)
- Asia/Novokuznetsk (UTC+ 07:00)
- Asia/Krasnoyarsk (UTC+ 07:00)
- Asia/Saigon (UTC+ 07:00)

- Asia/Yangon (UTC+ 06:30)
- Asia/Rangun (UTC+ 06:30)
- India/Cocos (UTC+ 06:30)
- Asia/Kashgar (UTC+ 06:00)
- DII/GMT-6 (UTC+ 06:00)
- Asia/Almaty (UTC+ 06:00)
- Asia/Dacca (UTC+ 06:00)
- Asia/Omsk (UTC+ 06:00)
- Asia/Dhaka (UTC+ 06:00)
- India/Chagos (UTC+ 06:00)
- Asia/Qyzylorda (UTC+ 06:00)
- Asia/Bishkek (UTC+ 06:00)
- Antartika/Vostok (UTC+ 06:00)
- Asia/Urumqi (UTC+ 06:00)
- Asia/Thimbu (UTC+ 06:00)
- Asia/Thimphu (UTC+ 06:00)
- Asia/Kathmandu (UTC+ 05:45)
- Asia/Katmandu (UTC+ 05:45)
- Asia/Kolkata (UTC+ 05:30)
- Asia/Kolombo (UTC+ 05:30)
- Asia/Kalkuta (UTC+ 05:30)
- Asia/Aqtau (UTC+ 05:00)
- DII/GMT-5 (UTC+ 05:00)
- Asia/Samarkand (UTC+ 05:00)
- Asia/Karachi (UTC+ 05:00)
- Asia/Yekaterinburg (UTC+ 05:00)
- Asia/Dushanbe (UTC+ 05:00)
- India/Maladewa (UTC+ 05:00)
- Asia/Lisan (UTC+ 05:00)

- Asia/Tashkent (UTC+ 05:00)
- Antartika/Mawson (UTC+ 05:00)
- Asia/Aqtobe (UTC+ 05:00)
- Asia/Ashkhabad (UTC+ 05:00)
- Asia/Ashgabat (UTC+ 05:00)
- Asia/Atyrau (UTC+ 05:00)
- India/Kerguelen (UTC+ 05:00)
- Iran (UTC+ 04:30)
- Asia/Teheran (UTC+ 04:30)
- Asia/Kabul (UTC+ 04:30)
- Asia/Yerevan (UTC+ 04:00)
- DII/GMT-4 (UTC+ 04:00)
- DII/GMT-4 (UTC+ 04:00)
- Asia/Dubai (UTC+ 04:00)
- India/Reuni (UTC+ 04:00)
- Eropa/Saratov (UTC+ 04:00)
- Eropa/Samara (UTC+ 04:00)
- India/Mahe (UTC+ 04:00)
- Asia/Baku (UTC+ 04:00)
- Asia/Muscat (UTC+ 04:00)
- Eropa/Volgograd (UTC+ 04:00)
- Eropa/Astrakhan (UTC+ 04:00)
- Asia/Tbilisi (UTC+ 04:00)
- Eropa/Ulyanovsk (UTC+ 04:00)
- Asia/Aden (UTC+ 03:00)
- Afrika/Nairobi (UTC+ 03:00)
- Eropa/Istanbul (UTC+ 03:00)
- DII/GMT-3 (UTC+ 03:00)
- Eropa/Zaporozhye (UTC+ 03:00)

- Israel (UTC+ 03:00)
- India/Comoro (UTC+ 03:00)
- Antartika/Syowa (UTC+ 03:00)
- Afrika/Mogadishu (UTC+ 03:00)
- Eropa/Bukares (UTC+ 03:00)
- Afrika/Asmera (UTC+ 03:00)
- Eropa/Mariehamn (UTC+ 03:00)
- Asia/Istanbul (UTC+ 03:00)
- Eropa/Tiraspol (UTC+ 03:00)
- Eropa/Moskow (UTC+ 03:00)
- Eropa/Chisinau (UTC+ 03:00)
- Eropa/Helsinki (UTC+ 03:00)
- Asia/Beirut (UTC+ 03:00)
- Asia/Tel_Aviv (UTC+ 03:00)
- Afrika/Djibouti (UTC+ 03:00)
- Eropa/Simferopol (UTC+ 03:00)
- Eropa/Sofia (UTC+ 03:00)
- Asia/Gaza (UTC+ 03:00)
- Afrika/Asmara (UTC+ 03:00)
- Eropa/Riga (UTC+ 03:00)
- Asia/Baghdad (UTC+ 03:00)
- Asia/Damaskus (UTC+ 03:00)
- Afrika/Dar_es_Salaam (UTC+ 03:00)
- Afrika/Addis_Ababa (UTC+ 03:00)
- Eropa/Uzhgorod (UTC+ 03:00)
- Asia/Yerusalem (UTC+ 03:00)
- Asia/Riyadh (UTC+ 03:00)
- Asia/Kuwait (UTC+ 03:00)
- Eropa/Kirov (UTC+ 03:00)

- Afrika/Kampala (UTC+ 03:00)
- Eropa/Minsk (UTC+ 03:00)
- Asia/Qatar (UTC+ 03:00)
- Eropa/Kiev (UTC+ 03:00)
- Asia/Bahrain (UTC+ 03:00)
- Eropa/Vilnius (UTC+ 03:00)
- India/Antananarivo (UTC+ 03:00)
- India/Mayotte (UTC+ 03:00)
- Eropa/Tallinn (UTC+ 03:00)
- Turki (UTC+ 03:00)
- Afrika/Juba (UTC+ 03:00)
- Asia/Nikosia (UTC+ 03:00)
- Asia/Keluarga (UTC+ 03:00)
- W-SU (UTC+ 03:00)
- MALAM (UTC+ 03:00)
- Asia/Hebron (UTC+ 03:00)
- Asia/Amman (UTC+ 03:00)
- Eropa/Nikosia (UTC+ 03:00)
- Eropa/Athina (UTC+ 03:00)
- Afrika/Kairo (UTC+ 02:00)
- Afrika/Mbabane (UTC+ 02:00)
- Eropa/Brussel (UTC+ 02:00)
- Eropa/Warsawa (UTC+ 02:00)
- CET (UTC+ 02:00)
- Eropa/Luksemburg (UTC+ 02:00)
- DII/GMT-2 (UTC+ 02:00)
- Libya (UTC+ 02:00)
- Afrika/Kigali (UTC+ 02:00)
- Afrika/Tripoli (UTC+ 02:00)

- Eropa/Kaliningrad (UTC+ 02:00)
- Afrika/Windhoek (UTC+ 02:00)
- Eropa/Malta (UTC+ 02:00)
- Eropa/Busingen (UTC+ 02:00)
-
- Eropa/Skopje (UTC+ 02:00)
- Eropa/Sarajevo (UTC+ 02:00)
- Eropa/Roma (UTC+ 02:00)
- Eropa/Zürich (UTC+ 02:00)
- Eropa/Gibraltar (UTC+ 02:00)
- Afrika/Lubumbashi (UTC+ 02:00)
- Eropa/Vaduz (UTC+ 02:00)
- Eropa/Ljubljana (UTC+ 02:00)
- Eropa/Berlin (UTC+ 02:00)
- Eropa/Stockholm (UTC+ 02:00)
- Eropa/Budapest (UTC+ 02:00)
- Eropa/Zagreb (UTC+ 02:00)
- Eropa/Paris (UTC+ 02:00)
- Afrika/Ceuta (UTC+ 02:00)
- Eropa/Praha (UTC+ 02:00)
- Antartika/Troll (UTC+ 02:00)
- Afrika/Gaborone (UTC+ 02:00)
- Eropa/Kopenhagen (UTC+ 02:00)
- Eropa/Wina (UTC+ 02:00)
- Eropa/Tirane (UTC+ 02:00)
- BERTEMU (UTC+ 02:00)
- Eropa/Amsterdam (UTC+ 02:00)
- Afrika/Maputo (UTC+ 02:00)
- Eropa/San_Marino (UTC+ 02:00)

- Polandia (UTC+ 02:00)
- Eropa/Andorra (UTC+ 02:00)
- Eropa/Oslo (UTC+ 02:00)
- Eropa/Podgorica (UTC+ 02:00)
- Afrika/Bujumbura (UTC+ 02:00)
- Atlantik/Jan_Mayen (UTC+ 02:00)
- Afrika/Maseru (UTC+ 02:00)
- Eropa/Madrid (UTC+ 02:00)
- Afrika/Blantyre (UTC+ 02:00)
- Afrika/Lusaka (UTC+ 02:00)
- Afrika/Harare (UTC+ 02:00)
- Afrika/Khartoum (UTC+ 02:00)
- Afrika/Johannesburg (UTC+ 02:00)
- Eropa/Beograd (UTC+ 02:00)
- Eropa/Bratislava (UTC+ 02:00)
- Arktik/Longyearbyen (UTC+ 02:00)
- Mesir (UTC+ 02:00)
- Eropa/Vatikan (UTC+ 02:00)
- Eropa/Monako (UTC+ 02:00)
- Eropa/London (UTC+ 01:00)
- DII/GMT-1 (UTC+ 01:00)
- Eropa/Jersey (UTC+ 01:00)
- Eropa/Guernsey (UTC+ 01:00)
- Eropa/Isle_of_man (UTC+ 01:00)
- Afrika/Tunis (UTC+ 01:00)
- Afrika/Malabo (UTC+ 01:00)
- GB-Eire (UTC+ 01:00)
- Afrika/Lagos (UTC+ 01:00)
- Afrika/Aljir (UTC+ 01:00)

- GB (UTC+ 01:00)
- Portugal (UTC+ 01:00)
- Afrika/Sao_Tome (UTC+ 01:00)
- Afrika/Ndjamena (UTC+ 01:00)
- Atlantik/Faeroe (UTC+ 01:00)
- Eire (UTC+ 01:00)
- Atlantik/Faroe (UTC+ 01:00)
- Eropa/Dublin (UTC+ 01:00)
- Afrika/Libreville (UTC+ 01:00)
- Afrika/El_Aaiun (UTC+ 01:00)
- Afrika/El_Aaiun (UTC+ 01:00)
- Afrika/Douala (UTC+ 01:00)
- Afrika/Brazzaville (UTC+ 01:00)
- Afrika/Porto-Novo (UTC+ 01:00)
- Atlantik/Madeira (UTC+ 01:00)
- Eropa/Lisbon (UTC+ 01:00)
- Atlantik/Kenari (UTC+ 01:00)
- Afrika/Casablanca (UTC+ 01:00)
- Eropa/Belfast (UTC+ 01:00)
- Afrika/Luanda (UTC+ 01:00)
- Afrika/Kinshasa (UTC+ 01:00)
- Afrika/Bangui (UTC+ 01:00)
- BASAH (UTC+ 01:00)
- Afrika/Niamey (UTC+ 01:00)
- GMT (UTC+ 00:00)
- DII/GMT-0 (UTC+ 00:00)
- Atlantik/St_Helena (UTC+ 00:00)
- DII/GMT+0 (UTC+ 00:00)
- Afrika/Banjul (UTC+ 00:00)

- DII/GMT (UTC+00:00)
- Afrika/Freetown (UTC+ 00:00)
- Afrika/Bamako (UTC+ 00:00)
- Afrika/Conakry (UTC+ 00:00)
- Universal (UTC+ 00:00)
- Afrika/Nouakchott (UTC+ 00:00)
- UTC (UTC+ 00:00)
- DII/Universal (UTC+00:00)
- Atlantik/Azores (UTC+ 00:00)
- Afrika/Abidjan (UTC+ 00:00)
- Afrika/Accra (UTC+ 00:00)
- DII/UCT (UTC+00:00)
- GMT0 (UTC+ 00:00)
- Zulu (UTC+ 00:00) Zulu (UTC+ 00:00)
- Afrika/Ouagadougou (UTC+ 00:00)
- Atlantik/Reykjavik (UTC+ 00:00)
- DII/Zulu (UTC+ 00:00)
- Islandia (UTC+ 00:00)
- Afrika/Lome (UTC+ 00:00)
- Greenwich (UTC+ 00:00)
- DII/GMT0 (UTC+ 00:00)
- Amerika/Danmarkshavn (UTC+ 00:00)
- Afrika/Dakar (UTC+ 00:00)
- Afrika/Bissau (UTC+ 00:00)
- DII/Greenwich (UTC+ 00:00)
- Afrika/Timbuktu (UTC+ 00:00)
- UCT (UTC+ 00:00)
- Afrika/Monrovia (UTC+ 00:00)
- DII/UTC (UTC+ 00:00)

Tutorial ekspresi rumus

Anda dapat mengikuti tutorial ini untuk menggunakan ekspresi rumus di AWS IoT SiteWise.

Topik

- [Gunakan string dalam rumus](#)
- [Filter titik data](#)
- [Hitung titik data yang cocok dengan kondisi](#)
- [Data terlambat dalam rumus](#)
- [Kualitas data dalam rumus](#)
- [Nilai tidak terdefinisi, tak terbatas, dan meluap](#)

Gunakan string dalam rumus

Anda dapat beroperasi pada string dalam ekspresi rumus Anda. Anda juga dapat memasukkan string dari variabel yang mereferensikan atribut dan properti pengukuran.

Important

Ekspresi rumus hanya dapat menampilkan nilai ganda atau string. Ekspresi bersarang dapat menampilkan tipe data lain, seperti string, tetapi rumus secara keseluruhan harus mengevaluasi ke angka atau string. Anda dapat menggunakan [fungsi jp](#) untuk mengonversi string ke angka. Nilai Boolean harus 1 (true) atau 0 (false). Untuk informasi selengkapnya, lihat [Nilai tidak terdefinisi, tak terbatas, dan meluap](#).

AWS IoT SiteWise menyediakan fitur ekspresi rumus berikut yang dapat Anda gunakan untuk beroperasi pada string:

- [String literal](#)
- [Operator indeks](#) (`s[index]`)
- [Operator irisan](#) (`s[start:end:step]`)
- [Fungsi perbandingan](#), yang dapat Anda gunakan membandingkan string berdasarkan urutan leksikografis
- [Fungsi string](#), yang mencakup `jp` fungsi yang dapat mengurai objek JSON serial dan mengonversi string menjadi angka

Filter titik data

Anda dapat menggunakan [fungsi if](#) untuk menyaring titik data yang tidak memenuhi kondisi.

`if` Fungsi mengevaluasi kondisi dan mengembalikan nilai yang berbeda untuk `true` dan `false` hasil. Anda dapat menggunakan [konstanta none](#) sebagai output untuk satu kasus `if` fungsi untuk membuang titik data untuk kasus itu.

Untuk memfilter titik data yang cocok dengan kondisi

- Buat transformasi yang menggunakan `if` fungsi untuk menentukan kondisi yang memeriksa apakah kondisi terpenuhi, dan kembali `none` sebagai `result_if_false` nilai `result_if_true` atau.

Example Contoh: Saring titik data di mana air tidak mendidih

Pertimbangkan skenario di mana Anda memiliki pengukuran, `temp_c`, yang menyediakan suhu (dalam Celcius) air dalam mesin. Anda dapat menentukan transformasi berikut untuk menyaring titik data di mana air tidak mendidih:

- Transform: `boiling_temps = if(gte(temp_c, 100), temp_c, none)` — Mengembalikan suhu jika lebih besar dari atau sama dengan 100 derajat Celcius, jika tidak mengembalikan tidak ada titik data.

Hitung titik data yang cocok dengan kondisi

Anda dapat menggunakan [fungsi perbandingan](#) dan [sum \(\)](#) untuk menghitung jumlah titik data yang kondisinya benar.

Untuk menghitung titik data yang cocok dengan suatu kondisi

1. Buat transformasi yang menggunakan fungsi perbandingan untuk menentukan kondisi filter pada properti lain.
2. Buat metrik yang menjumlahkan titik data di mana kondisi itu terpenuhi.

Example Contoh: Hitung jumlah titik data di mana air mendidih

Pertimbangkan skenario di mana Anda memiliki pengukuran, `temp_c`, yang menyediakan suhu (dalam Celcius) air dalam mesin. Anda dapat menentukan sifat transformasi dan metrik berikut untuk menghitung jumlah titik data tempat air mendidih:

- Transform: `is_boiling = gte(temp_c, 100)` — Kembali 1 jika suhu lebih besar dari atau sama dengan 100 derajat Celcius, jika tidak kembali 0.
- Metrik: `boiling_count = sum(is_boiling)` — Mengembalikan jumlah titik data di mana air mendidih.

Data terlambat dalam rumus

AWS IoT SiteWise mendukung keterlambatan konsumsi data yang berusia hingga 7 hari. Saat AWS IoT SiteWise menerima data terlambat, itu menghitung ulang nilai yang ada untuk metrik apa pun yang memasukkan data akhir di jendela sebelumnya. Penghitungan ulang ini menghasilkan biaya pemrosesan data.

Note

Saat AWS IoT SiteWise menghitung properti yang memasukkan data terlambat, ia menggunakan ekspresi rumus setiap properti saat ini.

Setelah AWS IoT SiteWise menghitung ulang jendela sebelumnya untuk metrik, itu menggantikan nilai sebelumnya untuk jendela itu. Jika Anda mengaktifkan notifikasi untuk metrik tersebut, AWS IoT SiteWise juga akan memancarkan notifikasi nilai properti. Ini berarti Anda dapat menerima pemberitahuan pembaruan nilai properti baru untuk properti dan stempel waktu yang sama yang sebelumnya Anda terima notifikasi. Jika aplikasi atau data lake Anda menggunakan notifikasi nilai properti, Anda harus memperbarui nilai sebelumnya dengan nilai baru sehingga datanya akurat.

Kualitas data dalam rumus

Di AWS IoT SiteWise, setiap titik data memiliki kode kualitas, yang dapat berupa salah satu dari yang berikut:

- GOOD— Data tidak terpengaruh oleh masalah apa pun.
- BAD— Data dipengaruhi oleh masalah seperti kegagalan sensor.
- UNCERTAIN— Data dipengaruhi oleh masalah seperti ketidakakuratan sensor.

AWS IoT SiteWise hanya mengkonsumsi data GOOD berkualitas ketika menghitung transformasi dan metrik. AWS IoT SiteWise hanya menghasilkan data GOOD berkualitas untuk perhitungan yang berhasil. Jika komputasi tidak berhasil, maka AWS IoT SiteWise tidak menampilkan titik data untuk

perhitungan itu. Hal ini dapat terjadi jika perhitungan menghasilkan nilai undefined, infinite, atau overflow.

Untuk informasi selengkapnya tentang cara menanyakan data dan memfilter berdasarkan kualitas data, lihat [Data kueri dari AWS IoT SiteWise](#).

Nilai tidak terdefinisi, tak terbatas, dan meluap

Beberapa ekspresi rumus (seperti $x / 0$, $\sqrt{-1}$, atau $\log(0)$) menghitung nilai yang tidak terdefinisi dalam sistem bilangan real, tak terbatas, atau di luar rentang yang didukung oleh AWS IoT SiteWise. Ketika ekspresi properti aset menghitung nilai undefined, infinite, atau overflow, AWS IoT SiteWise tidak menampilkan titik data untuk perhitungan tersebut.

AWS IoT SiteWise juga tidak menampilkan titik data jika menghitung nilai non-numerik sebagai hasil dari ekspresi rumus. Ini berarti bahwa jika Anda mendefinisikan rumus yang menghitung string, array, atau [konstanta none](#), maka AWS IoT SiteWise tidak menampilkan titik data untuk perhitungan itu.

Example Contoh

Masing-masing ekspresi rumus berikut menghasilkan nilai yang tidak AWS IoT SiteWise dapat mewakili sebagai angka. AWS IoT SiteWise tidak menampilkan titik data saat menghitung ekspresi rumus ini.

- $x / 0$ tidak terdefinisi.
- $\log(0)$ tidak terdefinisi.
- $\sqrt{-1}$ tidak terdefinisi dalam sistem bilangan real.
- "hello" + " world" adalah string.
- `jp({'values':[3,6,7]}, '$.values')` adalah sebuah array.
- `if(gte(temp, 300), temp, none)` temp adalah none kapan kurang dari 300.

Buat model komposit khusus (komponen)

Model komposit khusus, atau komponen jika Anda menggunakan konsol, menyediakan tingkat organisasi lain untuk model aset dan model komponen Anda. Anda dapat menggunakannya untuk menyusun model Anda dengan mengelompokkan properti atau mereferensikan model lain. Untuk informasi selengkapnya tentang bekerja dengan model komposit khusus, lihat [Model komposit khusus \(komponen\)](#).

Anda membuat model komposit khusus dalam model aset atau model komponen yang ada. Ada dua jenis model komposit khusus. Untuk mengelompokkan properti terkait dalam model, Anda dapat membuat model komposit kustom sebaris. Untuk mereferensikan model komponen dalam model aset atau model komponen, Anda dapat membuat model komposit component-model-based khusus.

Bagian berikut menjelaskan cara menggunakan AWS IoT SiteWise API untuk membuat model komposit khusus.

Topik

- [Buat komponen inline \(konsol\)](#)
- [Buat model komposit kustom inline \(AWS CLI\)](#)
- [Buat component-model-based komponen \(konsol\)](#)
- [Buat model komposit component-model-based khusus \(AWS CLI\)](#)

Buat komponen inline (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk membuat komponen inline yang mendefinisikan propertinya sendiri.

Note

Karena ini adalah komponen inline, properti ini hanya berlaku untuk model aset saat ini dan tidak dibagikan di tempat lain.

Jika Anda perlu menghasilkan model yang dapat digunakan kembali (misalnya, untuk berbagi di antara beberapa model aset, atau menyertakan beberapa instance dalam satu model aset), Anda harus membuat komponen berdasarkan model komponen. Lihat bagian berikut untuk detailnya.

Untuk membuat komponen (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset yang ingin Anda tambahkan komponennya.
4. Pada tab Properties, pilih Components.
5. Pilih Buat komponen.

6. Pada halaman Create component, lakukan hal berikut:
 - a. Masukkan Nama untuk komponen, seperti **ServoMotor** atau **ServoMotor Model1**. Nama ini harus unik di semua komponen di akun Anda di Wilayah ini.
 - b. (Opsional) Tambahkan definisi Atribut untuk model. Atribut mewakili informasi yang jarang berubah. Untuk informasi selengkapnya, lihat [Tentukan data statis \(atribut\)](#).
 - c. (Opsional) Tambahkan definisi Pengukuran untuk model. Pengukuran mewakili aliran data dari peralatan Anda. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).
 - d. (Opsional) Tambahkan definisi Transform untuk model. Transformasi adalah rumus yang memetakan data dari satu bentuk ke bentuk lainnya. Untuk informasi selengkapnya, lihat [Mengubah data \(mengubah\)](#).
 - e. (Opsional) Tambahkan definisi Metrik untuk model. Metrik adalah rumus yang mengumpulkan data dari interval waktu. Metrik dapat memasukkan data dari aset terkait, sehingga Anda dapat menghitung nilai yang mewakili operasi Anda atau subset operasi Anda. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).
 - f. Pilih Buat komponen.

Buat model komposit kustom inline (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat model komposit kustom inline yang mendefinisikan propertinya sendiri.

Gunakan [CreateAssetModelCompositeModel](#) operasi untuk membuat model inline dengan properti. Operasi ini mengharuskan muatan dengan struktur berikut.

Note

Karena ini adalah model komposit sebaris, properti ini hanya berlaku untuk model aset saat ini dan tidak dibagikan di tempat lain. Apa yang membuatnya “sebaris” adalah bahwa ia tidak memberikan nilai untuk `composedAssetModelId` bidang tersebut.

Jika Anda perlu menghasilkan model yang dapat digunakan kembali (misalnya, untuk berbagi di antara beberapa model aset, atau menyertakan beberapa contoh dalam satu model aset), Anda harus membuat model component-model-based komposit sebagai gantinya. Lihat bagian berikut untuk detailnya.

```
{
  "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"
    }
  ]
}
```

Buat component-model-based komponen (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk membuat komponen berdasarkan model komponen.

Untuk membuat component-model-based komponen (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset yang ingin Anda tambahkan komponennya.
4. Pada tab Properties, pilih Components.
5. Pilih Buat komponen.
6. Pada halaman Create component, lakukan hal berikut:
 - a. Pilih model komponen yang ingin Anda gunakan berdasarkan komponennya.
 - b. Masukkan Nama untuk komponen, seperti **ServoMotor** atau **ServoMotor Model**. Nama ini harus unik di semua komponen di akun Anda di Wilayah ini.

- c. Pilih Buat komponen.

Buat model komposit component-model-based khusus (AWS CLI)

Anda dapat menggunakan AWS CLI untuk membuat model komposit component-model-based khusus dalam model aset Anda. Model komposit component-model-based kustom adalah referensi ke model komponen yang telah Anda definisikan di tempat lain.

Gunakan [CreateAssetModelCompositeModel](#) operasi untuk membuat model komposit component-model-based khusus. Operasi ini mengharapkan muatan dengan struktur berikut.

Note

Dalam contoh ini, nilai `composedAssetModelId` adalah ID model aset atau ID eksternal dari model komponen yang ada. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna. Untuk contoh cara membuat model komponen, lihat [Buat model komponen \(AWS CLI\)](#).

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "composedAssetModelId": component model ID
}
```

Karena ini hanya referensi, model komposit component-model-based khusus tidak memiliki properti sendiri, selain nama.

Jika Anda ingin menambahkan beberapa contoh komponen yang sama ke model aset Anda (misalnya, mesin CNC yang memiliki beberapa motor servo), Anda dapat menambahkan beberapa model komposit component-model-based khusus yang masing-masing memiliki nama mereka sendiri tetapi yang semuanya mereferensikannya sama. `composedAssetModelId`

Anda dapat membuat sarang komponen di dalam komponen lain. Untuk melakukannya, Anda dapat menambahkan model component-model-based komposit, seperti yang ditunjukkan dalam contoh ini, ke salah satu model komponen Anda.

Buat aset untuk model aset di AWS IoT SiteWise

Anda dapat membuat aset dari model aset. Anda harus memiliki model aset sebelum Anda dapat membuat aset. Jika Anda belum membuat model aset, lihat [Buat model aset di AWS IoT SiteWise](#).

Note

Anda hanya dapat membuat aset dari ACTIVE model. Jika status model Anda tidak ACTIVE, Anda mungkin perlu menunggu hingga beberapa menit sebelum Anda dapat membuat aset dari model itu. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

Topik

- [Buat aset \(konsol\)](#)
- [Buat aset \(AWS CLI\)](#)
- [Konfigurasi aset baru](#)

Buat aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk membuat aset.

Untuk membuat aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih Buat aset.
4. Pada halaman Create asset, lakukan hal berikut:
 - a. Untuk Model, pilih model aset untuk membuat aset.

Note

Jika model Anda tidak AKTIF, Anda harus menunggu hingga aktif, atau menyelesaikan masalah jika gagal.

- b. Masukkan Nama untuk aset Anda.

- c. (Opsional) Tambahkan tag untuk aset Anda. Untuk informasi selengkapnya, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).
- d. Pilih Buat aset.

Saat Anda membuat aset, AWS IoT SiteWise konsol akan menavigasi ke halaman aset baru. Di halaman ini, Anda dapat melihat Status aset, yang awalnya DIBUAT. Halaman ini diperbarui secara otomatis, sehingga Anda dapat menunggu status aset diperbarui.

 Note

Proses pembuatan aset bisa memakan waktu hingga satu menit. Setelah Status AKTIF, Anda dapat melakukan operasi pembaruan pada aset Anda. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

Setelah Anda membuat aset, lihat [Konfigurasi aset baru](#).

Buat aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat aset dari model aset.

Anda harus memiliki `assetModelId` untuk membuat aset. Jika Anda membuat model aset, tetapi tidak mengetahuinya `assetModelId`, gunakan [ListAssetModels](#) API untuk melihat semua model aset Anda.

Untuk membuat aset dari model aset, gunakan [CreateAsset](#) API dengan parameter berikut:

- `assetName` Nama aset baru. Beri nama aset Anda untuk membantu Anda mengidentifikasinya.
- `assetModelId`— ID aset. Ini adalah ID aktual dalam format UUID, atau `externalId:myExternalId` jika memilikinya. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Untuk membuat aset (AWS CLI)

- Jalankan perintah berikut untuk membuat aset. Ganti *asset-name* dengan nama untuk aset dan *asset-model-id* dengan ID atau ID eksternal dari model aset.

```
aws iotsitewise create-asset \  
  --asset-name asset-name \  
  --asset-model-id asset-model-id
```

Operasi mengembalikan respons yang berisi detail dan status aset baru Anda dalam format berikut.

```
{  
  "assetId": "String",  
  "assetArn": "String",  
  "assetStatus": {  
    "state": "String",  
    "error": {  
      "code": "String",  
      "message": "String"  
    }  
  }  
}
```

Aset state adalah CREATING sampai aset dibuat.

Note

Proses pembuatan aset bisa memakan waktu hingga satu menit. Untuk memeriksa status aset Anda, gunakan [DescribeAsset](#) operasi dengan ID aset Anda sebagai `assetId` parameter. Setelah aset state selesai ACTIVE, Anda dapat melakukan operasi pembaruan pada aset Anda. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

Setelah Anda membuat aset, lihat [Konfigurasi aset baru](#).

Konfigurasi aset baru

Setelah membuat aset AWS IoT SiteWise, ada beberapa langkah selanjutnya yang dapat Anda ambil untuk sepenuhnya memanfaatkan aset dan datanya. Langkah-langkah ini mungkin termasuk mengonfigurasi aliran data untuk menelan data dari aset, menyiapkan alarm dan pemberitahuan untuk memantau kinerja aset, membuat visualisasi dan dasbor untuk menampilkan data aset, dan

mengintegrasikan aset dengan AWS layanan lain atau aplikasi pihak ketiga untuk analisis atau otomatisasi lebih lanjut.

Selesai mengonfigurasi aset Anda dengan tindakan opsional berikut:

- [Mengelola aliran data untuk AWS IoT SiteWise](#) jika aset Anda memiliki properti pengukuran.
- [Perbarui nilai atribut](#) jika aset Anda memiliki nilai atribut yang unik.
- [Mengasosiasikan dan memisahkan aset](#) jika aset Anda adalah aset induk.

Cari aset di Konsol AWS IoT SiteWise

Gunakan fungsionalitas Konsol AWS IoT SiteWise pencarian untuk menemukan aset berdasarkan metadata dan filter nilai properti real-time.

Prasyarat

AWS IoT SiteWise memerlukan izin untuk berintegrasi dengan AWS IoT TwinMaker mengatur, dan memodelkan data industri dengan lebih baik. Jika Anda telah memberikan izin AWS IoT SiteWise, gunakan [ExecuteQuery](#) API. Jika Anda belum memberikan izin untuk AWS IoT SiteWise, dan membutuhkan bantuan untuk memulai, lihat [Integrasikan AWS IoT SiteWise dan AWS IoT TwinMaker](#).

Pencarian lanjutan di Konsol AWS IoT SiteWise

Pencarian metadata

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Pencarian lanjutan di bawah Aset.
3. Di bawah Pencarian lanjutan pilih opsi pencarian Metadata.
4. Isi parameternya. Isi kolom sebanyak mungkin untuk pencarian yang efisien.
 - a. Nama aset — Masukkan nama aset lengkap, atau sebagian nama untuk pencarian luas.
 - b. Nama properti — Masukkan nama properti lengkap, atau sebagian nama untuk pencarian luas.
 - c. Operator — Pilih operator dari:
 - =
 - <

- >
 - <=
 - >=
- d. Nilai properti — Nilai ini dibandingkan dengan nilai terbaru properti.
- e. Jenis nilai properti — Tipe data properti. Pilih dari yang berikut ini:
- Ganda
 - bilangan bulat
 - Tali
 - Boolean
5. Pilih Cari.
6. Dari tabel Hasil pencarian, pilih aset dari kolom Nama. Ini membawa Anda ke halaman aset terperinci untuk aset itu.

The screenshot shows the AWS IoT SiteWise console interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, a search bar, and regional settings for 'N. Virginia'. Below this, the 'Assets' page is displayed. The page title is 'Assets', and there's a 'Create asset' button. A description states: '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.'

The 'Advanced search' section is active, showing a search query: 'Level-2' for Asset name, 'power_max' for Property name, and '20' for Property value, with the operator '>' and 'Double' for Property value type. A 'Search' button is highlighted. Below the search bar, the 'Search results (2)' are shown in a table:

Name	Asset id	Description
Level-2-asset-1	d0e9019b-9c38-4316-b574-38317aa38143	
Level-2-asset-2	b9c0d2fc-1527-42ce-8ba2-d1a4e8ff43de	Example description

Pencarian parsial

Semua parameter tidak perlu disediakan untuk pencarian aset. Berikut adalah beberapa contoh pencarian sebagian menggunakan opsi pencarian Metadata:

- Temukan aset dengan namanya:
 - Masukkan nilai di bidang Asset name saja.
 - Nama properti dan kolom nilai Properti kosong.
- Temukan aset yang berisi properti dengan nama tertentu:
 - Masukkan nilai di bidang Nama properti saja.
 - Bidang Asset name dan Property value kosong.
- Temukan aset berdasarkan nilai terbaru dari properti mereka:
 - Masukkan nilai di bidang Nama properti dan nilai Properti.
 - Pilih jenis nilai Operator dan Properti.

Pencarian pembuat kueri

1. Arahkan ke Konsol AWS IoT SiteWise.
2. Di panel navigasi, pilih Pencarian lanjutan di bawah Aset.
3. Di bawah Pencarian lanjutan pilih opsi Query builder.
4. Di panel Query builder, tulis kueri SQL Anda untuk mengambil, dan. `asset_name asset_id asset_description`
5. Pilih Cari.
6. Dari tabel Hasil pencarian, pilih aset dari kolom Nama. Ini membawa Anda ke halaman aset terperinci untuk aset itu.

The screenshot shows the AWS IoT SiteWise 'Assets' page. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a region dropdown set to 'N. Virginia'. Below the navigation bar, the page title 'Assets' is displayed with a 'Create asset' button. A descriptive paragraph explains that assets represent industrial devices and processes. The 'Advanced search' section is active, showing a 'Query builder' tab. The query entered is: `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`. Below the query builder, the 'Search results (2)' are shown in a table.

Name	Asset id	Description
Level-2a-asset-2	4fed596d-e903-4338-86db-34ca9301233a	Generator #3
Level-2b-asset-2	b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d	Generator #2

Note

- SELECTKlausa dalam kueri SQL harus menyertakan `asset_id` bidang `asset_name` dan untuk memastikan aset yang valid dalam tabel hasil Penelusuran.
- Pembuat Query hanya menampilkan Name, Asset id, dan Description dalam tabel hasil. Menambahkan lebih banyak bidang ke SELECT klausa tidak menambahkan lebih banyak kolom ke tabel hasil

Perbarui nilai atribut

Aset mewarisi atribut model aset mereka, termasuk nilai default atribut. Dalam beberapa kasus, Anda akan ingin menyimpan atribut default model aset, seperti untuk properti produsen aset. Dalam kasus lain, Anda akan ingin memperbarui atribut yang diwariskan, seperti untuk lintang dan bujur aset.

Updating an attribute value (console)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk memperbarui nilai properti aset atribut.

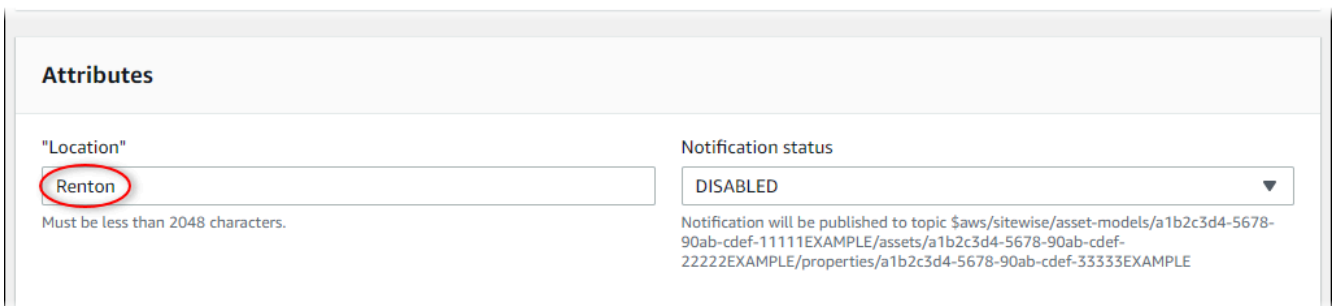
Untuk memperbarui nilai atribut (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda perbarui atributnya.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih Edit.
5. Temukan atribut yang akan diperbarui, lalu masukkan nilai barunya.



6. Pilih Simpan.

Updating an attribute value (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk memperbarui nilai atribut.

Anda harus mengetahui aset `assetId` dan properti Anda `propertyId` untuk menyelesaikan prosedur ini. Anda juga dapat menggunakan ID eksternal. Jika Anda membuat aset dan tidak mengetahuinya `assetId`, gunakan [ListAssets](#) API untuk mencantumkan semua aset untuk model tertentu. Gunakan [DescribeAsset](#) operasi untuk melihat properti aset Anda termasuk properti IDs.

Gunakan [BatchPutAssetPropertyValue](#) operasi untuk menetapkan nilai atribut ke aset Anda. Anda dapat menggunakan operasi ini untuk mengatur beberapa atribut sekaligus. Payload operasi ini berisi daftar entri, dan setiap entri berisi ID aset, ID properti, dan nilai atribut.

Untuk memperbarui nilai atribut (AWS CLI)

1. Buat file bernama `batch-put-payload.json` dan salin objek JSON berikut ke dalam file. Contoh payload ini menunjukkan cara mengatur lintang dan bujur turbin angin. Perbarui IDs, nilai, dan stempel waktu untuk memodifikasi payload untuk kasus penggunaan Anda.

```
{
  "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
          }
        }
      ]
    }
  ]
}
```

```
}
```

- Setiap entri dalam payload berisi `entryId` yang dapat Anda definisikan sebagai string unik apa pun. Jika ada entri permintaan yang gagal, setiap kesalahan akan berisi permintaan `entryId` yang sesuai sehingga Anda tahu permintaan mana yang harus dicoba lagi.
- Untuk menetapkan nilai atribut, Anda dapat menyertakan satu timestamp-quality-value (TQV) struktur dalam daftar `propertyValues` untuk setiap properti atribut. Struktur ini harus berisi yang baru `value` dan saat `initimestamp`.
 - `value`— Struktur yang berisi salah satu bidang berikut, tergantung pada jenis properti yang ditetapkan:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
 - `timestamp`— Struktur yang berisi waktu epoch Unix saat ini dalam hitungan detik, `timeInSeconds` AWS IoT SiteWise menolak setiap titik data dengan stempel waktu yang ada lebih dari 7 hari di masa lalu atau lebih baru dari 5 menit di masa depan.

Untuk informasi lebih lanjut tentang cara menyiapkan muatan [BatchPutAssetPropertyValue](#), lihat [Menelan data dengan AWS IoT SiteWise APIs](#).

2. Jalankan perintah berikut untuk mengirim nilai atribut ke AWS IoT SiteWise:

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

Mengasosiasikan dan memisahkan aset

Jika model aset Anda menentukan hierarki model aset anak, Anda dapat mengaitkan aset anak ke aset Anda. Aset induk dapat mengakses dan mengumpulkan data dari aset terkait. Untuk informasi selengkapnya tentang model aset hierarkis, lihat [Tentukan hierarki model aset](#).

Topik

- [Mengaitkan dan memisahkan aset \(konsol\)](#)

- [Mengasosiasikan dan memisahkan aset \(AWS CLI\)](#)

Mengaitkan dan memisahkan aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk mengaitkan dan memisahkan aset.

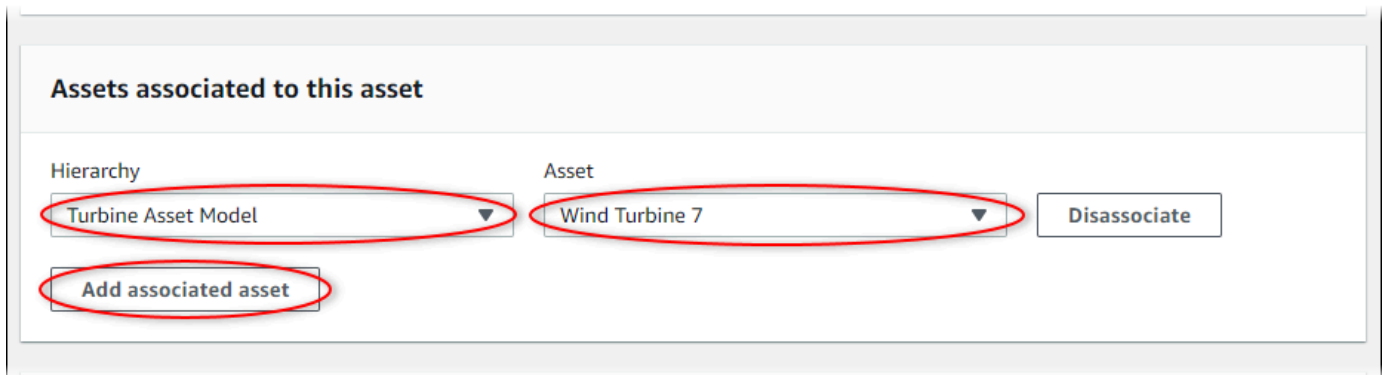
Untuk mengaitkan aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset induk yang ingin Anda kaitkan dengan aset anak.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih Edit.
5. Di Aset yang terkait dengan aset ini, pilih Tambahkan aset terkait.



6. Untuk Hierarki, pilih hierarki yang mendefinisikan hubungan antara aset induk dan aset anak.
7. Untuk Aset, pilih aset anak untuk diasosiasikan.
8. Pilih Simpan.

Untuk memisahkan aset (konsol)

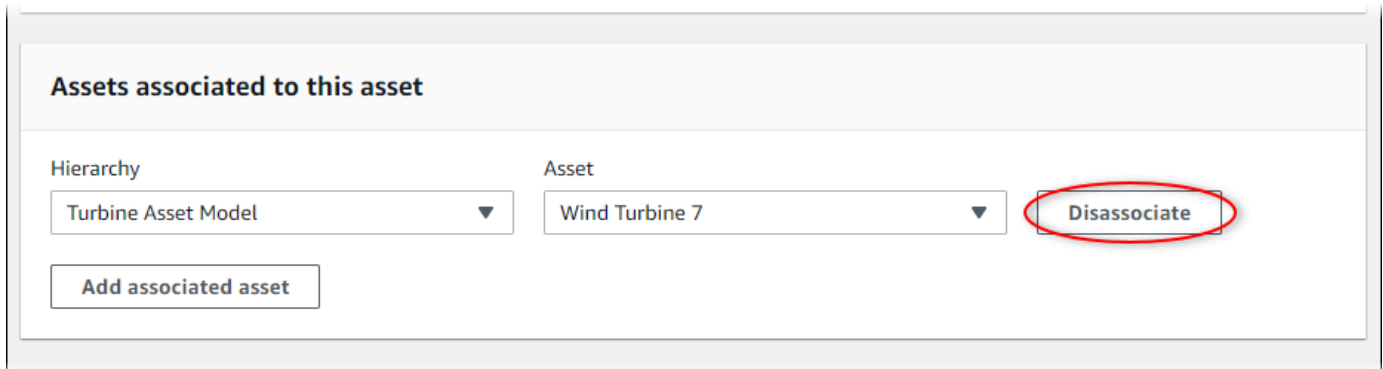
1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.

- Pilih aset induk yang ingin Anda pisahkan aset anak.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

- Pilih Edit.
- Dalam Aset yang terkait dengan aset ini, pilih Disassociate untuk aset tersebut.



Assets associated to this asset

Hierarchy	Asset	
Turbine Asset Model ▼	Wind Turbine 7 ▼	Disassociate

Add associated asset

- Pilih Simpan.

Mengasosiasikan dan memisahkan aset ()AWS CLI

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk mengaitkan dan memisahkan aset.

Untuk prosedur ini, Anda harus mengetahui ID hierarki (`hierarchyId`) dalam model aset induk yang mendefinisikan hubungan dengan model aset anak. Gunakan [DescribeAsset](#) operasi untuk menemukan ID hierarki dalam respons.

Untuk menemukan ID hierarki

- Jalankan perintah berikut untuk menggambarkan aset induk. Ganti *parent-asset-id* dengan ID aset induk atau ID eksternal.

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

Operasi mengembalikan respons yang berisi detail aset. Respons berisi `assetHierarchies` daftar yang memiliki struktur berikut:

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

ID hierarki adalah `id` nilai untuk hierarki dalam daftar hierarki aset.

Setelah Anda memiliki ID hierarki, Anda dapat mengaitkan atau memisahkan aset dengan hierarki tersebut.

Untuk mengaitkan aset anak dengan aset induk, gunakan [AssociateAssets](#) operasi. Untuk memisahkan aset anak dari aset induk, gunakan [DisassociateAssets](#) operasi. Tentukan parameter berikut, yang sama untuk kedua operasi:

- `assetId`— ID aset induk atau ID eksternal.
- `hierarchyId`— ID hierarki atau ID eksternal dalam aset induk.
- `childAssetId`— ID aset anak atau ID eksternal.

Untuk mengaitkan aset (AWS CLI)

- Jalankan perintah berikut untuk mengaitkan aset anak ke aset induk. Ganti *parent-asset-id* *hierarchy-id*, dan *child-asset-id* dengan masing-masing IDs:

```
aws iotsitewise associate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

Untuk memisahkan aset (AWS CLI)

- Jalankan perintah berikut untuk memisahkan aset anak dari aset induk. Ganti *parent-asset-id* hierarchy-id,, dan *child-asset-id* dengan masing-masing IDs:

```
aws iotsitewise disassociate-assets \  
  --asset-id parent-asset-id \  
  --hierarchy-id hierarchy-id \  
  --child-asset-id child-asset-id
```

Perbarui aset dan model

Anda dapat memperbarui aset, model aset, dan model komponen AWS IoT SiteWise untuk mengubah nama dan definisinya. Operasi pembaruan ini asinkron dan membutuhkan waktu untuk menyebar. AWS IoT SiteWise Periksa status aset atau model sebelum Anda membuat perubahan tambahan. Anda harus menunggu hingga perubahan menyebar sebelum Anda dapat terus menggunakan aset atau model yang diperbarui.

Topik

- [Perbarui aset di AWS IoT SiteWise](#)
- [Perbarui model aset dan model komponen](#)
- [Perbarui model komposit khusus \(komponen\)](#)
- [Penguncian optimis untuk penulisan model aset](#)

Perbarui aset di AWS IoT SiteWise

Anda dapat menggunakan AWS IoT SiteWise konsol atau API untuk memperbarui nama aset.

Saat Anda memperbarui aset, status aset adalah UPDATING hingga perubahan menyebar. Untuk informasi selengkapnya, lihat [Status aset dan model](#).

Topik

- [Memperbarui aset \(konsol\)](#)
- [Memperbarui aset \(AWS CLI\)](#)

Memperbarui aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk memperbarui detail aset.

Untuk memperbarui aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang akan diperbarui.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih Edit.
5. Perbarui Nama aset.
6. (Opsional) Di halaman ini, perbarui informasi lain untuk aset tersebut. Untuk informasi selengkapnya, lihat berikut ini:
 - [Mengelola aliran data untuk AWS IoT SiteWise](#)
 - [Perbarui nilai atribut](#)
 - [Berinteraksi dengan AWS layanan lain](#)
7. Pilih Simpan.

Memperbarui aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk memperbarui nama aset.

Gunakan [UpdateAsset](#) operasi untuk memperbarui aset. Tentukan parameter berikut:

- `assetId`— ID aset. Ini adalah ID aktual dalam format UUID, atau `externalId:myExternalId` jika memilikinya. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.
- `assetName` Nama baru aset tersebut.

Untuk memperbarui nama aset (AWS CLI)

- Jalankan perintah berikut untuk memperbarui nama aset. Ganti *asset-id* dengan ID atau ID eksternal aset. Perbarui *asset-name* dengan nama baru untuk aset tersebut.

```
aws iotsitewise update-asset \  
  --asset-id asset-id \  
  --asset-name asset-name
```

Perbarui model aset dan model komponen

Gunakan AWS IoT SiteWise konsol atau API untuk memperbarui model aset atau model komponen.

Anda tidak dapat mengubah tipe atau tipe data properti yang ada, atau jendela metrik yang ada. Anda juga tidak dapat mengubah jenis model dari model aset ke model komponen, atau sebaliknya.

Important

- Jika Anda menghapus properti dari model aset atau model komponen, AWS IoT SiteWise menghapus semua data sebelumnya untuk properti tersebut. Untuk model komponen, ini memengaruhi semua model aset yang menggunakan model komponen tersebut, jadi berhati-hatilah untuk memahami seberapa luas perubahan Anda dapat diterapkan.
- Jika Anda menghapus definisi hierarki dari model aset, AWS IoT SiteWise lepaskan semua aset dalam hierarki tersebut.

Saat Anda memperbarui model aset, setiap aset berdasarkan model tersebut mencerminkan perubahan apa pun yang Anda buat pada model yang mendasarinya. Sampai perubahan menyebar, setiap aset memiliki UPDATING negara. Anda harus menunggu sampai aset tersebut kembali ke ACTIVE negara sebelum Anda berinteraksi dengan mereka. Selama waktu ini, status model aset yang diperbarui akan menjadi PROPAGATING.

Saat Anda memperbarui model komponen, setiap model aset yang menggabungkan model komponen tersebut mencerminkan perubahan. Sampai perubahan model komponen menyebar, setiap model aset yang terpengaruh memiliki UPDATING status, diikuti oleh PROPAGATING saat memperbarui aset terkait, seperti yang dijelaskan dalam paragraf sebelumnya. Anda harus

menunggu sampai model aset tersebut kembali ke ACTIVE keadaan sebelum Anda berinteraksi dengannya. Selama waktu ini, status model komponen yang diperbarui akan menjadi PROPAGATING.

Untuk informasi selengkapnya, lihat [Status aset dan model](#).

Topik

- [Memperbarui model aset atau komponen \(konsol\)](#)
- [Memperbarui model aset atau komponen \(AWS CLI\)](#)

Memperbarui model aset atau komponen (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk memperbarui model aset atau model komponen.

Untuk memperbarui model aset atau model komponen (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset atau model komponen yang akan diperbarui.
4. Pilih Edit.
5. Pada halaman Edit model, lakukan salah satu hal berikut:
 - Dalam detail Model, ubah Nama model.
 - Ubah salah satu definisi Atribut. Anda tidak dapat mengubah tipe Data dari atribut yang ada. Untuk informasi selengkapnya, lihat [Tentukan data statis \(atribut\)](#).
 - Ubah salah satu definisi Pengukuran. Anda tidak dapat mengubah tipe Data pengukuran yang ada. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).
 - Ubah salah satu definisi Transform. Untuk informasi selengkapnya, lihat [Mengubah data \(mengubah\)](#).
 - Ubah salah satu definisi Metrik. Anda tidak dapat mengubah Interval waktu metrik yang ada. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).
 - (Hanya model aset) Ubah salah satu definisi Hierarki. Anda tidak dapat mengubah model Hierarki hierarki yang ada. Untuk informasi selengkapnya, lihat [Tentukan hierarki model aset](#).
6. Pilih Simpan.

 Note

Permintaan pembaruan yang dibuat di konsol ditolak, jika pengguna lain berhasil memperbarui model aset sejak Anda terakhir membuka halaman Edit model. Konsol meminta pengguna untuk Menyegarkan halaman Edit model, untuk mengambil model yang diperbarui. Anda harus melakukan pembaruan lagi, dan coba lagi penyimpanan Anda. Lihat [Penguncian optimis untuk penulisan model aset](#) untuk detail selengkapnya.

Memperbarui model aset atau komponen (AWS CLI)

Gunakan AWS Command Line Interface (AWS CLI) untuk memperbarui model aset atau model komponen.

Gunakan [UpdateAssetModel](#) API untuk memperbarui nama, deskripsi, dan properti model aset atau model komponen. Hanya untuk model aset, Anda dapat memperbarui hierarki. Tentukan parameter berikut:

- `assetModelId`— ID aset. Ini adalah ID aktual dalam format UUID, atau `externalId:myExternalId` jika memilikinya. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Tentukan model yang diperbarui di payload. Untuk mempelajari tentang format yang diharapkan dari model aset atau model komponen, lihat [Buat model aset di AWS IoT SiteWise](#).

 Warning

[UpdateAssetModel](#) API menimpa model yang ada dengan model yang Anda berikan di payload. Untuk menghindari penghapusan properti atau hierarki model Anda, Anda harus menyertakan properti IDs dan definisinya dalam payload model yang diperbarui. Untuk mempelajari cara menanyakan struktur model yang ada, lihat [DescribeAssetModel](#) operasinya.

 Note

Prosedur berikut hanya dapat memperbarui model komposit tipe AWS/ALARM. Jika Anda ingin memperbarui model CUSTOM komposit, gunakan [UpdateAssetModelCompositeModel](#) sebagai gantinya. Untuk informasi selengkapnya, lihat [Perbarui model komposit khusus \(komponen\)](#).

Untuk memperbarui model aset atau model komponen (AWS CLI)

1. Jalankan perintah berikut untuk mengambil definisi model yang ada. Ganti *asset-model-id* dengan ID atau ID eksternal model aset atau model komponen untuk diperbarui.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Perintah di atas mengembalikan definisi model yang sesuai dengan versi terbaru model.

Untuk kasus penggunaan di mana model aset berada dalam FAILED status, ambil definisi model yang valid yang sesuai dengan versi aktifnya untuk membuat permintaan pembaruan Anda. Lihat [Versi model aset](#) untuk detail. Jalankan perintah berikut untuk mengambil definisi model aktif:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

Operasi mengembalikan respons yang berisi detail model. Responsnya memiliki struktur sebagai berikut.

```
{
  "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"
},
"assetModelVersion": "String",
"eTag": "String"
}
```

Untuk informasi lebih lanjut, lihat [DescribeAssetModel](#) operasi.

2. Buat file bernama `update-asset-model.json` dan salin respons perintah sebelumnya ke dalam file.
3. Hapus pasangan nilai kunci berikut dari objek JSON di `update-asset-model.json`:
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCompositeModelSummaries`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
 - `assetModelType`
 - `assetModelVersion`
 - `eTag`

[UpdateAssetModel](#) Operasi mengharapkan muatan dengan struktur berikut:

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. Pada `update-asset-model.json`, lakukan langkah-langkah berikut:
 - Ubah nama model aset (`assetModelName`).

- Mengubah, menambah, atau menghapus deskripsi model aset (`assetModelDescription`).
 - Mengubah, menambah, atau menghapus properti model aset (`assetModelProperties`). Anda tidak dapat mengubah properti `dataType` yang ada atau metrik `window` yang ada. Untuk informasi selengkapnya, lihat [Tentukan properti data](#).
 - Mengubah, menambah, atau menghapus salah satu hierarki model aset (`assetModelHierarchies`). Anda tidak dapat mengubah hierarki `childAssetModelId` yang ada. Untuk informasi selengkapnya, lihat [Tentukan hierarki model aset](#).
 - Mengubah, menambah, atau menghapus salah satu model komposit model aset tipe AWS/ALARM (`assetModelCompositeModels`). Alarm memantau properti lain sehingga Anda dapat mengidentifikasi kapan peralatan atau proses memerlukan perhatian. Setiap definisi alarm adalah model komposit yang menstandarisasi serangkaian properti yang digunakan alarm. Untuk informasi selengkapnya, lihat [Pantau data dengan alarm di AWS IoT SiteWise](#) dan [Tentukan alarm pada model aset di AWS IoT SiteWise](#).
5. Jalankan perintah berikut untuk memperbarui model aset dengan definisi yang disimpan di `update-asset-model.json`. Ganti *asset-model-id* dengan ID model aset:

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://model-payload.json
```

Important

Ketika beberapa pengguna memperbarui model aset pada saat yang sama, perubahan pengguna mungkin secara tidak sengaja ditimpa oleh pengguna lain. Untuk mencegah hal ini, Anda harus menentukan permintaan pembaruan bersyarat. Lihat [Penguncian optimis untuk penulisan model aset](#).

Perbarui model komposit khusus (komponen)

Anda dapat menggunakan AWS IoT SiteWise API untuk memperbarui model komposit khusus, atau AWS IoT SiteWise konsol untuk memperbarui komponen.

Topik

- [Perbarui komponen \(konsol\)](#)

- [Perbarui model komposit khusus \(AWS CLI\)](#)

Perbarui komponen (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk memperbarui komponen.

Untuk memperbarui komponen (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset di mana komponennya berada.
4. Pada tab Properties, pilih Components.
5. Pilih komponen yang ingin Anda perbarui.
6. Pilih Edit.
7. Pada halaman Edit komponen, lakukan salah satu hal berikut:
 - Dalam detail Model, ubah Nama model.
 - Ubah salah satu definisi Atribut. Anda tidak dapat mengubah tipe Data dari atribut yang ada. Untuk informasi selengkapnya, lihat [Tentukan data statis \(atribut\)](#).
 - Ubah salah satu definisi Pengukuran. Anda tidak dapat mengubah tipe Data pengukuran yang ada. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#).
 - Ubah salah satu definisi Transform. Untuk informasi selengkapnya, lihat [Mengubah data \(mengubah\)](#).
 - Ubah salah satu definisi Metrik. Anda tidak dapat mengubah Interval waktu metrik yang ada. Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).
8. Pilih Simpan.

Perbarui model komposit khusus (AWS CLI)

Gunakan AWS Command Line Interface (AWS CLI) untuk memperbarui model komposit kustom.

Untuk memperbarui nama atau deskripsi, gunakan [UpdateAssetModelCompositeModel](#) operasi. Hanya untuk model komposit khusus sebaris, Anda juga dapat memperbarui properti. Anda tidak dapat memperbarui properti model komposit component-model-based khusus, karena model komponen yang direferensikan menyediakan properti terkait.

⚠ Important

Jika Anda menghapus properti dari model komposit kustom, AWS IoT SiteWise menghapus semua data sebelumnya untuk properti tersebut. Anda tidak dapat mengubah tipe atau tipe data properti yang ada.

Untuk mengganti properti model komposit yang ada dengan yang baru dengan yang samaname, lakukan hal berikut:

1. Kirim `UpdateAssetModelCompositeModel` permintaan dengan seluruh properti yang ada dihapus.
2. Kirim `UpdateAssetModelCompositeModel` permintaan kedua yang mencakup properti baru. Properti aset baru akan name sama dengan yang sebelumnya dan AWS IoT SiteWise akan menghasilkan unik baruid.

Untuk memperbarui model komposit kustom (AWS CLI)

1. Untuk mengambil definisi model komposit yang ada, jalankan perintah berikut. Ganti *composite-model-id* dengan ID atau ID eksternal model komposit khusus untuk diperbarui, dan *asset-model-id* dengan model aset yang terkait dengan model komposit khusus. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS IoT SiteWise .

a. Jalankan perintah di bawah ini:

```
aws iotsitewise describe-asset-model-composite-model \
--asset-model-composite-model-id composite-model-id \
--asset-model-id asset-model-id
```

- b. Perintah di atas mengembalikan definisi model komposit yang sesuai dengan versi terbaru model terkait. Untuk kasus penggunaan di mana model aset berada dalam FAILED status, ambil definisi model yang valid yang sesuai dengan versi aktifnya untuk membuat permintaan pembaruan Anda. Lihat [Versi model aset](#) untuk detail.

c. Jalankan perintah berikut untuk mengambil definisi model aktif:

```
aws iotsitewise describe-asset-model-composite-model \
--asset-model-composite-model-id composite-model-id \
--asset-model-id asset-model-id \
--asset-model-version ACTIVE
```

- d. Untuk informasi lebih lanjut, lihat [DescribeAssetModelCompositeModel](#) operasi.
2. Buat file bernama `update-custom-composite-model.json`, lalu salin respons perintah sebelumnya ke dalam file.
3. Hapus setiap pasangan kunci-nilai dari objek JSON `update-custom-composite-model.json` kecuali untuk bidang berikut:
 - `assetModelCompositeModelName`
 - `assetModelCompositeModelDescription` (jika ada)
 - `assetModelCompositeModelProperties` (jika ada)
4. Pada `update-custom-composite-model.json`, lakukan langkah-langkah berikut:
 - Ubah nilai `assetModelCompositeModelName`.
 - Tambah atau hapus `assetModelCompositeModelDescription`, atau ubah nilainya.
 - Hanya untuk model komposit kustom sebaris: Ubah, tambahkan, atau hapus properti model aset apa pun di `assetModelCompositeModelProperties`.

Untuk informasi selengkapnya tentang format yang diperlukan untuk file ini, lihat sintaks permintaan untuk [UpdateAssetModelCompositeModel](#).

5. Jalankan perintah berikut untuk memperbarui model komposit khusus dengan definisi yang disimpan di `update-custom-composite-model.json`. Ganti *composite-model-id* dengan ID model komposit, dan *asset-model-id* dengan ID model aset yang ada di dalamnya.

```
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
```

Important

Ketika beberapa pengguna memperbarui model aset pada saat yang sama, perubahan pengguna mungkin secara tidak sengaja ditimpa oleh pengguna lain. Untuk mencegah hal ini, Anda harus menentukan permintaan pembaruan bersyarat. Lihat [Penguncian optimis untuk penulisan model aset](#).

Penguncian optimis untuk penulisan model aset

Saat memperbarui model aset, pengguna melakukan hal berikut:

1. Baca definisi model aset saat ini.
2. Edit definisi model aset dengan perubahan yang diperlukan.
3. Perbarui model aset dengan definisi baru.

Dalam skenario dengan dua pengguna memperbarui model, berikut ini dimungkinkan:

- Pengguna A membaca definisi model aset X.
- Pengguna B membaca definisi model aset X dan melakukan perubahan, memodifikasi definisi X.
- Pengguna A melakukan dan menimpa perubahan yang dibuat oleh pengguna B untuk model aset X, tanpa memverifikasi atau memasukkan perubahan Pengguna B.

Penguncian optimis adalah mekanisme yang digunakan oleh AWS IoT SiteWise untuk mencegah penimpaan yang tidak disengaja seperti skenario di atas. Penguncian optimis adalah strategi untuk memastikan versi model aset saat ini diperbarui atau dihapus, sama dengan versi saat ini di AWS IoT SiteWise. Ini melindungi penulisan model aset agar tidak ditimpa oleh pembaruan yang tidak disengaja.

Ikuti langkah-langkah ini untuk melakukan penulisan model aset dengan penguncian optimis:

Topik

- [Melakukan penulisan model aset dengan kunci optimis \(konsol\)](#)
- [Melakukan penulisan model aset dengan optimistic lock \(\)AWS CLI](#)

Melakukan penulisan model aset dengan kunci optimis (konsol)

Prosedur di bawah ini menjelaskan cara melakukan penulisan model aset dengan kunci optimis pada versi aktif model aset di konsol.

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset atau model komponen yang akan diperbarui.
4. Pilih Edit.

5. Buat perubahan pada halaman Edit model.
6. Pilih Simpan.

 Note

Terkadang, satu atau lebih pembaruan model yang berhasil terjadi antara saat pengguna mulai mengedit model, dan menyimpan pengeditan yang dibuat ke model.

Untuk memastikan pengguna tidak sengaja menimpa pembaruan baru yang berhasil, tulisan pengguna ditolak. Konsol menonaktifkan tombol Simpan, dan meminta pengguna untuk menyegarkan halaman Edit model. Pengguna harus memperbarui versi aktif baru dari model lagi. Pengguna harus melakukan langkah-langkah tambahan berikut:

7. Pilih Refresh.
8. Ikuti langkah 5 dan 6 lagi.

Melakukan penulisan model aset dengan optimistic lock ()AWS CLI

Prosedur di bawah ini menjelaskan cara melakukan penulisan model aset dengan penguncian optimis di. AWS CLI

1. Ambil yang ETag terkait dengan definisi model saat ini

ETag adalah token unik yang dihasilkan untuk setiap representasi baru dari model aset. Panggil [DescribeAssetModel](#) API untuk mengambil definisi model aset saat ini, dan terkait ETag dari respons.

Selama pembaruan bersamaan, pengguna melakukan pembaruan yang berhasil (model dalam ACTIVE status), atau pembaruan yang gagal (model dalam FAILED status). Untuk memastikan bahwa pengguna tidak sengaja menimpa pembaruan yang berhasil, Anda harus mengambil versi aktif model aset dari [Versi model aset](#), dan mendapatkan nilainya. ETag

Jalankan perintah berikut:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id \
--asset-model-version ACTIVE
```

Respons mengembalikan struktur berikut:

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  ...
  "eTag": "String"
}
```

Note

Anda harus mengambil versi terbaru dari model aset dan itu ETag agar tidak menimpa pembaruan apa pun.

2. Lakukan operasi UPDATE dan DELETE dengan kondisi tulis

Model aset berikut APIs mendukung penguncian optimis:

- [UpdateAssetModel](#)
- [DeleteAssetModel](#)
- [CreateAssetModelCompositeModel](#)
- [UpdateAssetModelCompositeModel](#)
- [DeleteAssetModelCompositeModel](#)

Note

Skenario di bawah ini menggunakan UpdateAssetModel API sebagai referensi. Ketentuan berlaku untuk semua operasi yang tercantum di atas.

Skenario di bawah ini menjelaskan kondisi penulisan yang berbeda tergantung pada persyaratan kontrol konkurensi:

- Jalankan perintah berikut agar tidak menimpa pembaruan yang berhasil. Versi aktif baru tidak boleh ada, karena versi aktif baca terakhir. Ganti e-tag dengan yang ETag dikembalikan dalam operasi API yang digunakan dalam pembacaan versi aktif.


```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match e-tag \  
  --match-for-version-type ACTIVE \  
  --cli-input-json file://model-payload.json
```

- Ketika pembuatan model gagal, versi aktif belum ada untuknya, karena dalam FAILED keadaan. Masih dimungkinkan untuk menerima versi aktif baru yang ada, sebelum perubahan Anda dilakukan. Jalankan perintah berikut untuk tidak menerima versi aktif baru, ketika versi aktif tidak ada selama pembacaan terakhir Anda.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-none-match "*" \  
  --match-for-version-type ACTIVE \  
  --cli-input-json file://model-payload.json
```

- Jalankan perintah berikut untuk menghindari menerima pembaruan yang berhasil atau tidak berhasil. Perintah ini mendefinisikan kondisi tulis yang memastikan bahwa versi terbaru tidak dibuat sejak versi terbaru Anda dibaca terakhir. Ganti e-tag dengan yang ETag dikembalikan dalam operasi API yang digunakan dalam pembacaan versi aktif.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match eTag \  
  --match-for-version-type LATEST \  
  --cli-input-json file://model-payload.json
```

Jika kondisi tulis dievaluasi FALSE, permintaan tulis gagal dengan.
`PreconditionFailedException`

Hapus aset dan model di AWS IoT SiteWise

Anda dapat menghapus aset dan model dari AWS IoT SiteWise saat Anda selesai menggunakannya. Operasi penghapusan tidak sinkron dan membutuhkan waktu untuk menyebar. AWS IoT SiteWise

Topik

- [Hapus aset di AWS IoT SiteWise](#)

- [Hapus model aset di AWS IoT SiteWise](#)

Hapus aset di AWS IoT SiteWise

Anda dapat menggunakan AWS IoT SiteWise konsol atau API untuk menghapus aset yang tidak lagi diperlukan di lingkungan Anda. Menghapus model aset juga menghapus semua aset terkait dan model komponen. Namun, penting untuk dicatat bahwa menghapus aset atau model adalah tindakan permanen, dan data apa pun yang terkait dengan sumber daya yang dihapus juga akan dihapus. Sebelum menghapus aset atau model, disarankan untuk meninjau dependensi atau integrasi apa pun yang mungkin terpengaruh dan memastikan bahwa Anda memiliki cadangan data penting apa pun.

Sebelum Anda dapat menghapus aset, Anda harus terlebih dahulu memisahkan aset turunannya dan memisahkannya dari aset induknya. Untuk informasi selengkapnya, lihat [Mengasosiasikan dan memisahkan aset](#). Jika Anda menggunakan AWS Command Line Interface (AWS CLI), Anda dapat menggunakan [ListAssociatedAssets](#) operasi untuk membuat daftar anak-anak aset.

Saat Anda menghapus aset, statusnya DELETING sampai perubahan menyebar. Untuk informasi selengkapnya, lihat [Status aset dan model](#). Setelah aset dihapus, Anda tidak dapat menanyakan aset tersebut. Jika Anda melakukannya, API mengembalikan respons HTTP 404.

Important

AWS IoT SiteWise menghapus semua data properti untuk aset yang dihapus.

Topik

- [Menghapus aset \(konsol\)](#)
- [Menghapus aset \(AWS CLI\)](#)

Menghapus aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menghapus aset.

Untuk menghapus aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.

3. Pilih aset yang akan dihapus.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Jika aset memiliki aset Terkait, hapus setiap aset. Anda dapat memilih nama aset untuk menavigasi ke halamannya, tempat Anda dapat menghapusnya.
5. Pada halaman aset, pilih Hapus.
6. Di kotak dialog Hapus aset, lakukan hal berikut:
 - a. Masukkan **Delete** untuk mengonfirmasi penghapusan.
 - b. Pilih Hapus.

Menghapus aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menghapus aset.

Gunakan [DeleteAsset](#) operasi untuk menghapus aset. Tentukan parameter berikut:

- `assetId`— ID aset. Ini adalah ID aktual dalam format UUID, atau `externalId:myExternalId` jika memilikinya. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Untuk menghapus aset (AWS CLI)

1. Jalankan perintah berikut untuk membuat daftar hierarki aset. Ganti `asset-id` dengan ID atau ID eksternal aset:

```
aws iotsitewise describe-asset --asset-id asset-id
```

Operasi mengembalikan respons yang berisi detail aset. Respons berisi `assetHierarchies` daftar yang memiliki struktur berikut:

```
{  
  ...  
  "assetHierarchies": [  
    ...  
  ]  
}
```

```
{
  "id": "String",
  "name": "String"
},
...
}
```

Untuk informasi lebih lanjut, lihat [DescribeAsset](#) operasi.

2. Untuk setiap hierarki, jalankan perintah berikut untuk mencantumkan turunan aset yang terkait dengan hierarki tersebut. Ganti *asset-id* dengan ID atau ID eksternal aset dan *hierarchy-id* dengan ID atau ID eksternal hierarki.

```
aws iotsitewise list-associated-assets \
  --asset-id asset-id \
  --hierarchy-id hierarchy-id
```

Untuk informasi lebih lanjut, lihat [ListAssociatedAssets](#) operasi.

3. Jalankan perintah berikut untuk menghapus setiap aset terkait dan kemudian menghapus aset tersebut. Ganti *asset-id* dengan ID atau ID eksternal aset.

```
aws iotsitewise delete-asset --asset-id asset-id
```

Hapus model aset di AWS IoT SiteWise

Anda dapat menggunakan AWS IoT SiteWise konsol atau API untuk menghapus model aset.

Sebelum Anda dapat menghapus model aset, Anda harus terlebih dahulu menghapus semua aset yang dibuat dari model aset.

Saat Anda menghapus model aset, statusnya DELETING sampai perubahan menyebar. Untuk informasi selengkapnya, lihat [Status aset dan model](#). Setelah model aset dihapus, Anda tidak dapat menanyakan model aset tersebut. Jika Anda melakukannya, API mengembalikan respons HTTP 404.

Topik

- [Menghapus model aset \(konsol\)](#)
- [Menghapus model aset \(AWS CLI\)](#)

Menghapus model aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menghapus model aset.

Untuk menghapus model aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset yang akan dihapus.
4. Jika model memiliki Aset, hapus setiap aset. Pilih nama aset untuk menavigasi ke halamannya, tempat Anda dapat menghapusnya. Untuk informasi selengkapnya, lihat [Menghapus aset \(konsol\)](#).
5. Pada halaman model, pilih Hapus.
6. Dalam kotak dialog Delete model, lakukan hal berikut:
 - a. Masukkan **Delete** untuk mengonfirmasi penghapusan.
 - b. Pilih Hapus.

Menghapus model aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menghapus model aset.

Gunakan [DeleteAssetModel](#) operasi untuk menghapus model aset. Tentukan parameter berikut:

- `assetModelId`— ID aset. Ini adalah ID aktual dalam format UUID, atau `externalId:myExternalId` jika memilikinya. Untuk informasi selengkapnya, lihat [Referensi objek dengan eksternal IDs](#) di AWS IoT SiteWise Panduan Pengguna.

Untuk menghapus model aset (AWS CLI)

1. Jalankan perintah berikut untuk mencantumkan semua aset yang dibuat dari model. Ganti *asset-model-id* dengan ID atau ID eksternal model aset.

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

Untuk informasi lebih lanjut, lihat [ListAssets](#) operasi.

2. Jika perintah sebelumnya mengembalikan aset apa pun dari model, hapus setiap aset. Untuk informasi selengkapnya, lihat [Menghapus aset \(AWS CLI\)](#).
3. Jalankan perintah berikut untuk menghapus model aset. Ganti *asset-model-id* dengan ID atau ID eksternal model aset.

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

Important

Untuk menghindari penghapusan model aset yang diperbarui secara bersamaan sejak operasi baca terakhir, Anda harus menentukan permintaan penghapusan bersyarat. Lihat [Penguncian optimis untuk penulisan model aset](#).

Operasi massal dengan aset dan model

Untuk bekerja dengan sejumlah besar aset atau model aset, gunakan operasi massal untuk mengimpor dan mengeksport sumber daya secara massal ke lokasi yang berbeda. Misalnya, Anda dapat membuat file data yang mendefinisikan aset atau model aset di bucket Amazon S3, dan menggunakan impor massal untuk membuat atau memperbaruinya. AWS IoT SiteWise Atau, jika Anda memiliki sejumlah besar aset atau model aset AWS IoT SiteWise, Anda dapat mengeksportnya ke Amazon S3.

Note

Anda melakukan operasi massal AWS IoT SiteWise dengan memanggil operasi di AWS IoT TwinMaker API. Anda dapat melakukan ini tanpa menyiapkan AWS IoT TwinMaker atau membuat AWS IoT TwinMaker ruang kerja. Yang Anda butuhkan hanyalah ember Amazon S3 tempat Anda dapat menempatkan konten Anda AWS IoT SiteWise .

Topik

- [Konsep dan terminologi kunci](#)
- [Fungsionalitas yang didukung](#)
- [Prasyarat operasi massal](#)

- [Jalankan pekerjaan impor massal](#)
- [Jalankan pekerjaan ekspor massal](#)
- [Pelacakan kemajuan pekerjaan dan penanganan kesalahan](#)
- [Impor contoh metadata](#)
- [Contoh metadata ekspor](#)
- [AWS IoT SiteWise skema pekerjaan transfer metadata](#)

Konsep dan terminologi kunci

AWS IoT SiteWise fitur impor dan ekspor massal bergantung pada konsep dan terminologi berikut:

- **Impor:** Tindakan memindahkan aset atau model aset dari file di bucket Amazon S3 ke file. AWS IoT SiteWise
- **Ekspor:** Tindakan memindahkan aset atau model aset dari AWS IoT SiteWise ke bucket Amazon S3.
- **Sumber:** Lokasi awal dari mana Anda ingin memindahkan konten.

Misalnya, bucket Amazon S3 adalah sumber impor, dan AWS IoT SiteWise merupakan sumber ekspor.

- **Tujuan:** Lokasi yang diinginkan di mana Anda ingin memindahkan konten Anda.

Misalnya, bucket Amazon S3 adalah tujuan ekspor, dan AWS IoT SiteWise merupakan tujuan impor.

- **AWS IoT SiteWise Skema:** Skema ini digunakan untuk mengimpor dan mengekspor metadata dari. AWS IoT SiteWise
- **Sumber daya tingkat atas:** AWS IoT SiteWise Sumber daya yang dapat Anda buat atau perbarui secara individual, seperti model aset atau aset.
- **Sub-sumber daya:** Sumber AWS IoT SiteWise daya bersarang dalam sumber daya tingkat atas. Contohnya termasuk properti, hierarki, dan model komposit.
- **Metadata:** Informasi penting yang diperlukan untuk mengimpor atau mengekspor sumber daya dengan sukses. Contoh metadata adalah definisi aset dan model aset.
- **metadataTransferJob:** Objek yang dibuat saat Anda menjalankan `CreateMetadataTransferJob`.

Fungsionalitas yang didukung

Topik ini menjelaskan apa yang dapat Anda lakukan ketika Anda menjalankan operasi massal. Operasi massal mendukung fungsionalitas berikut:

- Pembuatan sumber daya tingkat atas: Saat Anda mengimpor aset atau model aset yang tidak menentukan ID, atau ID yang tidak cocok dengan ID yang sudah ada, maka ID tersebut akan dibuat sebagai sumber daya baru.
- Penggantian sumber daya tingkat atas: Ketika Anda mengimpor aset atau model aset yang ID-nya cocok dengan yang sudah ada, maka itu akan menggantikan sumber daya yang ada.
- Pembuatan, penggantian, atau penghapusan subsumber daya: Ketika impor Anda menggantikan sumber daya tingkat atas seperti aset atau model aset, definisi baru menggantikan semua sub-sumber daya, seperti properti, hierarki, atau model komposit.

Misalnya, jika Anda memperbarui model aset selama impor massal, dan versi yang diperbarui mendefinisikan properti yang tidak ada pada aslinya, maka properti baru akan dibuat. Jika mendefinisikan properti yang sudah ada, maka properti yang ada akan diperbarui. Jika model aset yang diperbarui menghilangkan properti yang ada pada aslinya, maka properti tersebut akan dihapus.

- Tidak ada penghapusan sumber daya tingkat atas: Operasi massal tidak menghapus aset atau model aset. Operasi massal hanya membuat atau memperbaruinya.

Prasyarat operasi massal

Bagian ini menjelaskan prasyarat operasi massal, termasuk izin AWS Identity and Access Management (IAM) untuk bertukar sumber daya antara layanan dan mesin lokal Anda. AWS Sebelum Anda memulai operasi massal, selesaikan prasyarat berikut:

- Buat bucket Amazon S3 untuk menyimpan sumber daya. Untuk informasi selengkapnya tentang menggunakan Amazon S3, lihat [Apa itu Amazon S3?](#)

Izin IAM

Untuk melakukan operasi massal, Anda harus membuat kebijakan AWS Identity and Access Management (IAM) dengan izin yang memungkinkan pertukaran AWS sumber daya antara Amazon S3 AWS IoT SiteWise, dan mesin lokal Anda. Untuk informasi selengkapnya tentang membuat kebijakan IAM, lihat [Membuat kebijakan IAM](#).

Untuk melakukan operasi massal, Anda memerlukan kebijakan berikut.

AWS IoT SiteWise kebijakan

Kebijakan ini memungkinkan akses ke tindakan AWS IoT SiteWise API yang diperlukan untuk operasi massal:

```
{
  "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 kebijakan

Kebijakan ini memungkinkan akses ke operasi AWS IoT TwinMaker API yang Anda gunakan untuk bekerja dengan operasi massal:

```
{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
  "Resource": "*"
}
```

Kebijakan Amazon S3

Kebijakan ini menyediakan akses ke bucket Amazon S3 untuk mentransfer metadata untuk operasi massal.

For a specific Amazon S3 bucket

Jika Anda menggunakan satu bucket tertentu untuk bekerja dengan metadata operasi massal, kebijakan ini menyediakan akses ke bucket tersebut:

```
{
  "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

Jika Anda akan menggunakan banyak bucket berbeda untuk bekerja dengan metadata operasi massal, kebijakan ini menyediakan akses ke bucket apa pun:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}
```

Untuk informasi tentang pemecahan masalah operasi impor dan ekspor, lihat. [Memecahkan masalah impor dan ekspor massal](#)

Jalankan pekerjaan impor massal

Impor massal adalah tindakan memindahkan metadata ke ruang kerja. AWS IoT SiteWise Misalnya, impor massal dapat memindahkan metadata dari file lokal, atau file di bucket Amazon S3, ke ruang kerja. AWS IoT SiteWise

Langkah 1: Siapkan file untuk diimpor

Unduh file format AWS IoT SiteWise asli untuk mengimpor aset dan model aset. Lihat [AWS IoT SiteWise skema pekerjaan transfer metadata](#) untuk detail selengkapnya.

Langkah 2: Unggah file yang disiapkan ke Amazon S3

Unggah file ke Amazon S3. Lihat [Mengunggah file ke Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon untuk detailnya.

Impor metadata (konsol)

Anda dapat menggunakan metadata impor massal. Konsol AWS IoT SiteWise Ikuti [Langkah 1: Siapkan file untuk diimpor](#) dan [Langkah 2: Unggah file yang disiapkan ke Amazon S3](#) siapkan file yang siap diimpor.

Impor data dari Amazon S3 ke Konsol AWS IoT SiteWise

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Pilih Operasi massal Baru dari panel navigasi.
3. Pilih Impor baru untuk memulai proses impor.
4. Pada halaman metadata Impor:
 - Pilih Jelajahi Amazon S3 untuk melihat bucket dan file Amazon S3.
 - Arahkan ke bucket Amazon S3 yang berisi file impor yang disiapkan.
 - Pilih file yang akan diimpor.
 - Tinjau file yang dipilih, dan pilih Impor.
5. Operasi massal pada halaman SiteWise metadata Konsol AWS IoT SiteWise menampilkan pekerjaan impor yang baru dibuat dalam tabel kemajuan Pekerjaan.

Impor metadata ()AWS CLI

Untuk melakukan tindakan impor, gunakan prosedur berikut:

Impor data dari Amazon S3 ke AWS CLI

1. Buat file metadata yang menentukan sumber daya yang ingin Anda impor, mengikuti [AWS IoT SiteWise skema pekerjaan transfer metadata](#) Simpan file ini di bucket Amazon S3 Anda.

Untuk contoh file metadata yang akan diimpor, lihat [Impor contoh metadata](#)

2. Sekarang buat file JSON dengan badan permintaan. Badan permintaan menentukan sumber dan tujuan untuk pekerjaan transfer. File ini terpisah dari file dari langkah sebelumnya. Pastikan untuk menentukan bucket Amazon S3 Anda sebagai sumber dan iotsitewise sebagai tujuan.

Contoh berikut menunjukkan badan permintaan:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
```

```
{
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket/your_import_metadata.json"
    }
  }],
  "destination": {
    "type": "iotsitewise"
  }
}
```

3. Memanggil CreateMetadataTransferJob dengan menjalankan AWS CLI perintah berikut. Dalam contoh ini, file isi permintaan dari langkah sebelumnya diberi nama createMetadataTransferJobExport.json.

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://createMetadataTransferJobImport.json
```

Ini akan membuat pekerjaan transfer metadata, dan memulai proses mentransfer sumber daya yang Anda pilih.

Jalankan pekerjaan ekspor massal

Ekspor massal adalah tindakan memindahkan metadata dari AWS IoT SiteWise ruang kerja ke bucket Amazon S3.

Saat melakukan ekspor massal AWS IoT SiteWise konten ke Amazon S3, Anda dapat menentukan filter untuk membatasi model aset dan aset tertentu yang ingin Anda ekspor.

Filter harus ditentukan di `iotSiteWiseConfiguration` bagian dalam bagian sumber permintaan JSON Anda.

Note

Anda dapat menyertakan beberapa filter dalam permintaan Anda. Operasi massal akan mengekspor model aset dan aset yang cocok dengan filter mana pun.

Jika Anda tidak memberikan filter apa pun, operasi massal mengekspor semua model aset dan aset Anda.

Example minta badan dengan filter

```
{
  "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:::amzn-s3-demo-bucket"
    }
  }
}
```

Ekspor metadata (konsol)

Prosedur berikut menjelaskan tindakan ekspor konsol:

Buat pekerjaan ekspor di Konsol AWS IoT SiteWise

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Pilih Operasi massal Baru dari panel navigasi.
3. Pilih Ekspor baru untuk memulai proses ekspor.
4. Pada halaman metadata Ekspor:
 - Masukkan nama untuk pekerjaan ekspor. Ini adalah nama yang digunakan untuk file yang diekspor di bucket Amazon S3 Anda.
 - Pilih sumber daya Anda untuk diekspor, yang menetapkan filter untuk pekerjaan:
 - Ekspor semua aset dan model aset. Gunakan filter pada aset dan model aset.
 - Aset ekspor. Filter aset Anda.
 - Pilih aset yang akan digunakan untuk filter ekspor.
 - (Opsional) Tambahkan keturunan atau model aset terkait.
 - Model aset ekspor. Filter pada model aset Anda.
 - Pilih model aset yang akan digunakan untuk filter ekspor.
 - (Opsional) Tambahkan keturunan, atau aset terkait atau keduanya.
 - Pilih Berikutnya.
 - Arahkan ke bucket Amazon S3:
 - Pilih Jelajahi Amazon S3 untuk melihat bucket dan file Amazon S3.
 - Arahkan ke bucket Amazon S3 tempat file harus ditempatkan.
 - Pilih Berikutnya.
 - Tinjau pekerjaan ekspor dan pilih Ekspor.
5. Operasi massal pada halaman SiteWise metadata Konsol AWS IoT SiteWise menampilkan pekerjaan impor yang baru dibuat dalam tabel kemajuan Pekerjaan.

Untuk berbagai cara menggunakan filter saat mengekspor metadata, lihat. [Contoh metadata ekspor](#)

Metadata ekspor ()AWS CLI

Prosedur berikut menjelaskan tindakan AWS CLI ekspor:

Ekspor data dari AWS IoT SiteWise ke Amazon S3

1. Buat file JSON dengan badan permintaan Anda. Badan permintaan menentukan sumber dan tujuan untuk pekerjaan transfer. Contoh berikut menunjukkan badan permintaan contoh:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
  }],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Pastikan untuk menentukan bucket Amazon S3 Anda sebagai tujuan pekerjaan transfer metadata.

Note

Contoh ini akan mengekspor semua model aset dan aset Anda. Untuk membatasi ekspor ke model aset atau aset tertentu, Anda dapat menyertakan filter di badan permintaan Anda. Untuk informasi selengkapnya tentang menerapkan filter ekspor, lihat [Contoh metadata ekspor](#).

2. Simpan file isi permintaan Anda untuk digunakan pada langkah berikutnya. Dalam contoh ini, file dinamai `createMetadataTransferJobExport.json`.
3. Memanggil `CreateMetadataTransferJob` dengan menjalankan AWS CLI perintah berikut:

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://createMetadataTransferJobExport.json
```

Ganti file JSON input `createMetadataTransferJobExport.json` dengan nama file transfer Anda sendiri.

Pelacakan kemajuan pekerjaan dan penanganan kesalahan

Pekerjaan proses massal membutuhkan waktu untuk diproses. Setiap pekerjaan diproses dalam urutan AWS IoT SiteWise menerima permintaan. Ini diproses one-at-a-time untuk setiap akun. Ketika pekerjaan selesai, antrian berikutnya secara otomatis mulai diproses. AWS IoT SiteWise menyelesaikan pekerjaan secara asinkron dan memperbarui status masing-masing saat berlangsung. Setiap pekerjaan memiliki bidang status yang berisi status sumber daya dan pesan kesalahan, jika berlaku.

Negara dapat menjadi salah satu dari nilai berikut:

- **VALIDATING**— Memvalidasi pekerjaan termasuk format file yang dikirimkan, dan isinya.
- **PENDING**— Pekerjaan dalam antrian. Anda dapat membatalkan pekerjaan di negara bagian ini dari AWS IoT SiteWise konsol, tetapi semua negara bagian lain akan berlanjut hingga akhir.
- **RUNNING**— Memproses pekerjaan. Ini membuat dan memperbarui sumber daya seperti yang didefinisikan oleh file impor, atau mengeksport sumber daya berdasarkan filter pekerjaan ekspor yang dipilih. Jika dibatalkan, sumber daya apa pun yang diimpor oleh pekerjaan ini tidak akan dihapus. Untuk informasi selengkapnya, lihat [Tinjau kemajuan dan detail pekerjaan \(konsol\)](#).
- **CANCELLING**— Pekerjaan secara aktif dibatalkan.
- **ERRORS** Satu atau lebih sumber daya gagal diproses. Periksa laporan pekerjaan terperinci untuk informasi lebih lanjut. Untuk informasi selengkapnya, lihat [Periksa detail kesalahan \(konsol\)](#).
- **COMPLETED**— Job selesai tanpa kesalahan.
- **CANCELLED**— Pekerjaan dibatalkan dan tidak antri. Jika Anda membatalkan **RUNNING** pekerjaan, sumber daya yang sudah diimpor oleh pekerjaan ini pada saat pembatalan tidak dihapus. AWS IoT SiteWise

Topik

- [Pelacakan kemajuan pekerjaan](#)
- [Memeriksa kesalahan untuk AWS IoT SiteWise](#)

Pelacakan kemajuan pekerjaan

Tinjau kemajuan dan detail pekerjaan (konsol)

Lihat [Impor metadata \(konsol\)](#) atau [Eksport metadata \(konsol\)](#) untuk memulai pekerjaan massal.

Ikhtisar kemajuan pekerjaan di AWS IoT SiteWise konsol:

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Pilih Operasi massal Baru dari panel navigasi.
3. Tabel kemajuan Pekerjaan di AWS IoT SiteWise konsol, menampilkan daftar pekerjaan operasi massal.
4. Kolom tipe Job menjelaskan apakah itu pekerjaan ekspor atau impor. Kolom Tanggal yang diimpor menampilkan tanggal pekerjaan dimulai.
5. Kolom Status menampilkan status pekerjaan. Anda dapat memilih pekerjaan untuk melihat detail tentang pekerjaan itu.
6. Pekerjaan yang dipilih menunjukkan Sukses setelah berhasil, atau daftar kegagalan jika pekerjaan gagal. Deskripsi kesalahan juga ditampilkan dengan setiap jenis sumber daya.

Ikhtisar detail pekerjaan di AWS IoT SiteWise konsol:

Tabel kemajuan Pekerjaan di AWS IoT SiteWise konsol, menampilkan daftar pekerjaan operasi massal.

1. Pilih pekerjaan untuk melihat detail lebih lanjut.
2. Untuk pekerjaan impor, Data source ARN mewakili lokasi Amazon S3 dari file impor.
3. Untuk pekerjaan ekspor, Data destination ARN mewakili lokasi Amazon S3 file setelah ekspor.
4. StatusDanStatus reason, memberikan rincian tambahan tentang pekerjaan saat ini. Lihat [Pelacakan kemajuan pekerjaan dan penanganan kesalahan](#) untuk detail selengkapnya.
5. Queued positionIni mewakili posisi pekerjaan dalam antrian proses. Pekerjaan diproses satu per satu. Posisi antrian 1, menunjukkan bahwa pekerjaan akan diproses selanjutnya.
6. Halaman detail pekerjaan juga menampilkan jumlah kemajuan pekerjaan.
 - Jenis jumlah kemajuan pekerjaan adalah:
 - i. Total resources— Menunjukkan jumlah total aset dalam proses transfer.
 - ii. Succeeded— Menunjukkan jumlah aset yang berhasil ditransfer selama proses.
 - iii. Failed— Menunjukkan jumlah aset yang gagal selama proses.
 - iv. Skipped— Menunjukkan jumlah aset yang dilewati selama proses.

7. Status pekerjaan PENDING atau VALIDATING, menampilkan semua kemajuan pekerjaan dihitung sebagai-. Ini menunjukkan bahwa jumlah kemajuan pekerjaan sedang dievaluasi.
8. Status pekerjaan RUNNING menampilkan Total resources hitungan, pekerjaan yang diajukan untuk diproses. Hitungan terperinci (Succeeded, Failed, dan Skipped), berlaku untuk sumber daya yang diproses. Jumlah dari hitungan rinci lebih kecil dari Total resources hitungan, sampai status pekerjaan adalah COMPLETED atau ERROR.
9. Jika status pekerjaan adalah COMPLETED atau ERROR, Total resources hitungannya sama dengan jumlah hitungan rinci (Succeeded, Failed, dan Skipped).
10. Jika status pekerjaan adalah ERROR, periksa tabel Kegagalan Job untuk rincian tentang kesalahan dan kegagalan tertentu. Lihat [Periksa detail kesalahan \(konsol\)](#) untuk detail selengkapnya.

Tinjau kemajuan dan detail pekerjaan (AWS CLI)

Setelah memulai operasi massal, Anda dapat memeriksa atau memperbarui statusnya menggunakan tindakan API berikut:

- Untuk mengambil informasi tentang pekerjaan tertentu, gunakan tindakan [GetMetadataTransferJobAPI](#).

Mengambil informasi dengan **GetMetadataTransferJob** API:

1. Buat dan jalankan pekerjaan transfer. Panggil GetMetadataTransferJob API.

Example AWS CLI perintah:

```
aws iottwinmaker get-metadata-transfer-job \
  --metadata-transfer-job-id your_metadata_transfer_job_id \
  --region your_region
```

2. GetMetadataTransferJobAPI mengembalikan MetadataTransferJobProgress objek dengan parameter berikut:
 - SucceededCount - Menunjukkan jumlah aset yang berhasil ditransfer dalam proses.
 - FailedCount - Menunjukkan jumlah aset yang gagal selama proses.
 - SkippedCount - Menunjukkan jumlah aset yang dilewati selama proses.
 - TotalCount — Menunjukkan jumlah total aset dalam proses transfer.

Parameter ini menunjukkan status kemajuan pekerjaan. Jika statusnya `RUNNING`, mereka membantu melacak jumlah sumber daya yang masih akan diproses.

Jika Anda menemukan kesalahan validasi skema, atau jika `failedCount` lebih besar dari atau sama dengan 1, status kemajuan pekerjaan berubah menjadi `ERROR`. Laporan kesalahan lengkap untuk pekerjaan tersebut ditempatkan di bucket Amazon S3 Anda. Lihat [Memeriksa kesalahan untuk AWS IoT SiteWise](#) untuk detail selengkapnya.

- Untuk membuat daftar pekerjaan saat ini, gunakan tindakan [ListMetadataTransferJobs](#) API.

Gunakan file JSON untuk memfilter pekerjaan yang dikembalikan berdasarkan statusnya saat ini. Lihat prosedur berikut:

1. Untuk menentukan filter yang ingin Anda gunakan, buat file JSON AWS CLI masukan. ingin menggunakan:

```
{
  "sourceType": "s3",
  "destinationType": "iottwinmaker",
  "filters": [{
    "state": "COMPLETED"
  }]
}
```

Untuk daftar state nilai yang valid, lihat [ListMetadataTransferJobsFilter](#) di Panduan Referensi AWS IoT TwinMaker API.

2. Gunakan file JSON sebagai argumen dalam AWS CLI contoh perintah berikut:

```
aws iottwinmaker list-metadata-transfer-job --region your_region \
  --cli-input-json file://ListMetadataTransferJobsExample.json
```

- Untuk membatalkan pekerjaan, gunakan tindakan [CancelMetadataTransferJob](#) API. API ini membatalkan tugas transfer metadata tertentu, tanpa memengaruhi sumber daya apa pun yang sudah diekspor atau diimpor:

```
aws iottwinmaker cancel-metadata-transfer-job \
  --region your_region \
```

```
--metadata-transfer-job-id job-to-cancel-id
```

Memeriksa kesalahan untuk AWS IoT SiteWise

Periksa detail kesalahan (konsol)

Detail kesalahan di AWS IoT SiteWise konsol:

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Lihat tabel kemajuan Pekerjaan Konsol AWS IoT SiteWise untuk daftar pekerjaan operasi massal.
3. Pilih pekerjaan untuk melihat detail pekerjaan.
4. Jika status pekerjaan adalah COMPLETED atau ERROR, Total resources hitungannya sama dengan jumlah hitungan rinci (Succeeded, Failed, dan Skipped).
5. Jika status pekerjaan adalah ERROR, periksa tabel Kegagalan Job untuk rincian tentang kesalahan dan kegagalan tertentu.
6. Tabel kegagalan Job menampilkan konten dari laporan pekerjaan. Resource type Bidang menunjukkan lokasi kesalahan atau kegagalan, seperti berikut ini:
 - Misalnya, kesalahan validasi Bulk operations template di Resource type bidang menunjukkan bahwa templat impor dan format file skema metadata tidak cocok. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise skema pekerjaan transfer metadata](#).
 - Gagal Asset di Resource type lapangan menunjukkan bahwa aset tidak dibuat karena konflik dengan aset lain. Lihat [Kesalahan umum](#) untuk informasi tentang kesalahan dan konflik AWS IoT SiteWise sumber daya.

Periksa detail kesalahan (AWS CLI)

Untuk menangani dan mendiagnosis kesalahan yang dihasilkan selama tugas transfer, lihat prosedur berikut tentang penggunaan tindakan GetMetadataTransferJob API:

1. Setelah membuat dan menjalankan pekerjaan transfer, hubungi [GetMetadataTransferJob](#):

```
aws iottwinmaker get-metadata-transfer-job \  
  --metadata-transfer-job-id your_metadata_transfer_job_id \  
  --region us-east-1
```

2. Setelah Anda melihat status pekerjaan beralih keCOMPLETED, Anda dapat mulai memverifikasi hasil pekerjaan.
3. Ketika Anda memanggil `GetMetadataTransferJob`, ia mengembalikan objek yang dipanggil [`MetadataTransferJobProgress`](#).

`MetadataTransferJobProgress` Objek berisi parameter berikut:

- `FailedCount`: Menunjukkan jumlah aset yang gagal selama proses transfer.
 - `SkippedCount`: Menunjukkan jumlah aset yang dilewati selama proses transfer.
 - `SucceededCount`: Menunjukkan jumlah aset yang berhasil selama proses transfer.
 - `TotalCount`: Menunjukkan jumlah total aset yang terlibat dalam proses transfer.
4. Selain itu, panggilan API mengembalikan elemen `reportUrl`, yang berisi URL presigned. Jika pekerjaan transfer Anda memiliki masalah yang perlu Anda selidiki lebih lanjut, kunjungi url ini.

Impor contoh metadata

Bagian ini menunjukkan cara membuat file metadata untuk mengimpor model aset dan aset dengan operasi impor massal tunggal.

Contoh impor massal

Anda dapat mengimpor banyak model aset dan aset dengan satu operasi impor massal. Contoh berikut menunjukkan cara membuat file metadata untuk melakukan hal ini.

Dalam skenario contoh ini, Anda memiliki berbagai situs kerja yang berisi robot industri dalam sel kerja.

Contoh mendefinisikan dua model aset:

- `RobotModel1`: Model aset ini mewakili jenis robot tertentu yang Anda miliki di lokasi kerja Anda. Robot memiliki properti pengukuran, `Temperature`.
- `WorkCell1`: Model aset ini mewakili kumpulan robot dalam salah satu situs kerja Anda. Model aset mendefinisikan hierarki, `robotHierarchyOEM1`, untuk mewakili hubungan bahwa sel kerja berisi robot.

Contoh ini juga mendefinisikan beberapa aset:

- `WorkCell1`: sel kerja dalam situs Boston Anda

- RobotArm123456: robot di dalam sel kerja itu
- RobotArm987654: robot lain di dalam sel kerja itu

File metadata JSON berikut mendefinisikan model dan aset aset ini. Menjalankan impor massal dengan metadata ini menciptakan model aset dan aset di dalamnya AWS IoT SiteWise, termasuk hubungan hierarkisnya.

File metadata untuk impor

```
{
  "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"
        }
      ]
    }
  ]
}

```

Contoh on-boarding awal model dan aset

Dalam skenario contoh ini, Anda memiliki berbagai situs kerja yang berisi robot industri di sebuah perusahaan.

Contoh mendefinisikan beberapa model aset:

- **Sample_Enterprise**— Model aset ini mewakili perusahaan yang menjadi bagian dari situs tersebut. Model aset mendefinisikan hierarki, **Enterprise to Site**, untuk mewakili hubungan situs dengan perusahaan.

- **Sample_Site**— Model aset ini mewakili lokasi manufaktur di dalam perusahaan. Model aset mendefinisikan hierarki, **Site to Line**, untuk mewakili hubungan garis ke situs.
- **Sample_Welding Line**— Model aset ini mewakili jalur perakitan di dalam lokasi kerja. Model aset mendefinisikan hierarki, **Line to Robot**, untuk mewakili hubungan robot dengan garis.
- **Sample_Welding Robot**— Model aset ini mewakili jenis robot tertentu di lokasi kerja Anda.

Contoh ini juga mendefinisikan aset berdasarkan model aset.

- **Sample_AnyCompany Motor** Aset ini dibuat dari model **Sample_Enterprise** aset.
- **Sample_Chicago** Aset ini dibuat dari model **Sample_Site** aset.
- **Sample_Welding Line 1** Aset ini dibuat dari model **Sample_Welding Line** aset.
- **Sample_Welding Robot 1** Aset ini dibuat dari model **Sample_Welding Robot** aset.
- **Sample_Welding Robot 2** Aset ini dibuat dari model **Sample_Welding Robot** aset.

File metadata JSON berikut mendefinisikan model dan aset aset ini. Menjalankan impor massal dengan metadata ini menciptakan model aset dan aset di dalamnya AWS IoT SiteWise, termasuk hubungan hierarkisnya.

File JSON ke aset dan model onboard untuk impor

```
{
  "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"
      }
    ]
  }
]
}

```

Screenshot berikut adalah model yang ditampilkan Konsol AWS IoT SiteWise setelah Anda menjalankan contoh kode sebelumnya.

IoT SiteWise > Models

Models (4) Refresh Create component model Create asset model

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.

Name	Status	Model type	Date created	Date modified
Sample_Enterprise	ACTIVE	Asset model	November 10, 2023 at 11:22:13 (UT...)	November 10, 202...
Sample_Site	ACTIVE	Asset model	November 10, 2023 at 11:21:57 (UT...)	November 10, 202...
Sample_Welding Line	ACTIVE	Asset model	November 10, 2023 at 11:21:40 (UT...)	November 10, 202...
Sample_Welding Robot	ACTIVE	Asset model	November 10, 2023 at 11:21:24 (UT...)	November 10, 202...

Screenshot berikut adalah model, aset, dan hierarki yang ditampilkan di Konsol AWS IoT SiteWise setelah Anda menjalankan contoh kode sebelumnya.

IoT SiteWise > Assets

Assets (1) Refresh Create asset

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.

Filter top level assets

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-5:00)
Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-5:00)
Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-5:00)
Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-5:00)
Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-5:00)

Contoh orientasi aset tambahan

Contoh ini mendefinisikan aset tambahan untuk diimpor ke model aset yang ada di akun Anda:

- Sample_Welding Line 2 Aset ini dibuat dari model Sample_Welding Line aset.
- Sample_Welding Robot 3 Aset ini dibuat dari model Sample_Welding Robot aset.
- Sample_Welding Robot 4 Aset ini dibuat dari model Sample_Welding Robot aset.

Untuk membuat aset awal untuk contoh ini, lihat [Contoh on-boarding awal model dan aset](#).

File metadata JSON berikut mendefinisikan model dan aset aset ini. Menjalankan impor massal dengan metadata ini menciptakan model aset dan aset di dalamnya AWS IoT SiteWise, termasuk hubungan hierarkisnya.

File JSON ke aset tambahan onboard

```
{
  "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"
        }
    ]
}
]
}
}

```

Screenshot berikut adalah model, aset, dan hierarki yang ditampilkan di Konsol AWS IoT SiteWise setelah Anda menjalankan contoh kode sebelumnya.

IoT SiteWise > Assets

Assets (1) Refresh Create asset

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.

Filter top level assets

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)

Contoh orientasi properti baru

Contoh ini mendefinisikan properti baru pada model aset yang ada. Lihat [Contoh orientasi aset tambahan](#) untuk memasukkan aset dan model tambahan.

- **Joint 1 Temperature**— Properti ini ditambahkan ke model Sample_Welding Robot aset. Properti baru ini juga akan menyebar ke setiap aset yang dibuat dari model Sample_Welding Robot aset.

Untuk menambahkan properti baru ke model aset yang ada, lihat contoh file metadata JSON berikut. Seperti yang ditunjukkan dalam JSON, seluruh definisi model Sample_Welding Robot aset yang ada harus disediakan bersama dengan properti baru. Jika seluruh daftar properti dari definisi yang ada tidak disediakan, AWS IoT SiteWise menghapus properti yang dihilangkan.

File JSON ke properti baru onboard

Contoh ini menambahkan properti baru Joint 1 Temperature ke model aset.

```
{
  "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"
}
]
}

```

Contoh mengelola aliran data

Contoh ini menunjukkan dua cara mengelola aliran data yang terkait dengan properti aset. Saat mengganti nama alias properti aset, ada dua opsi untuk data historis yang saat ini disimpan dalam aliran data properti aset.

- Opsi satu - Simpan aliran data saat ini dan ganti nama alias saja, memungkinkan data historis dapat diakses dengan alias baru.

Dalam contoh file metadata JSON, properti aset dengan ID `External_Id_Welding_Robot_Cycle_Count` mengubah aliasnya menjadi `AnyCompany/Chicago/Welding Line/S3000/Count-Updated`. Data historis untuk properti aset ini tetap sama setelah perubahan ini.

- Opsi dua — Tetapkan aliran data baru ke properti aset yang dapat diakses dengan alias baru. Aliran data lama bersama dengan data historisnya masih dapat diakses dengan alias lama, tetapi tidak terkait dengan properti aset apa pun.

Dalam contoh file metadata JSON, properti aset dengan ID `External_Id_Welding_Robot_Joint_1_Current` mengubah aliasnya menjadi `AnyCompany/Chicago/Welding Line/S4999/1/Current`. Kali ini nilai tambahan `retainDataOnAliasChange` hadir dan diatur ke `False`. Dengan pengaturan ini, aliran data asli dipisahkan dari properti aset, dan aliran data baru dibuat tanpa data historis.

Untuk mengakses aliran data lama dengan data historis asli, di AWS Console Home, buka halaman Data Streams dan cari alias `AnyCompany/Chicago/Welding Line/S3000/1/Current` lama.

File JSON untuk memperbarui alias properti

```
{
  "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-Updated"
    },
    {
      "externalId": "External_Id_Welding_Robot_Joint_1_Current",
      "alias": "AnyCompany/Chicago/Welding Line/S4999/1/Current",
      "retainDataOnAliasChange": "FALSE"
    }
  ]
}
```

Contoh metadata ekspor

Saat melakukan ekspor massal AWS IoT SiteWise konten ke Amazon S3, Anda dapat menentukan filter untuk membatasi model aset dan aset tertentu yang ingin Anda ekspor.

Anda menentukan filter di `iotSiteWiseConfiguration` bagian dalam `sources` bagian badan permintaan Anda.

Note

Anda dapat menyertakan beberapa filter. Operasi massal akan mengekspor model aset atau aset apa pun yang cocok dengan filter mana pun.

Jika Anda tidak memberikan filter apa pun, maka operasi akan mengekspor semua model aset dan aset Anda.

```
{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [{
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [{
        list of filters
      }]
    }
  ]],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Filter berdasarkan model aset

Anda dapat memfilter model aset tertentu. Anda juga dapat menyertakan semua aset menggunakan model tersebut, atau semua model aset dalam hierarki. Anda tidak dapat menyertakan aset dan hierarki.

Untuk informasi selengkapnya tentang hierarki, lihat [Tentukan hierarki model aset](#).

Asset model

Filter ini mencakup model aset yang ditentukan:

```
"filterByAssetModel": {
  "assetModelId": "asset model ID"
}
```

```
}
```

Asset model and its assets

Filter ini mencakup model aset yang ditentukan, bersama dengan semua aset yang menggunakan model aset tersebut:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeAssets": true  
}
```

Asset model and its hierarchy

Filter ini mencakup model aset yang ditentukan, bersama dengan semua model aset terkait dalam hierarki:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeOffspring": true  
}
```

Filter berdasarkan aset

Anda dapat memfilter aset tertentu. Anda juga dapat menyertakan model asetnya, atau semua aset terkait dalam hierarki. Anda tidak dapat menyertakan model aset dan hierarki.

Untuk informasi selengkapnya tentang hierarki, lihat [Tentukan hierarki model aset](#).

Asset

Filter ini mencakup aset yang ditentukan:

```
"filterByAsset": {  
  "assetId": "asset ID"  
}
```

Asset and its asset model

Filter ini mencakup aset yang ditentukan, bersama dengan model aset yang digunakannya:

```
"filterByAsset": {
  "assetId": "asset ID",
  "includeAssetModel": true
}
```

Asset and its hierarchy

Filter ini mencakup aset yang ditentukan, bersama dengan semua aset terkait dalam hierarki:

```
"filterByAsset": {
  "assetId": "asset ID",
  "includeOffspring": true
}
```

AWS IoT SiteWise skema pekerjaan transfer metadata

Gunakan skema pekerjaan transfer AWS IoT SiteWise metadata untuk referensi saat melakukan operasi impor dan ekspor massal Anda sendiri:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTSiteWise",
  "description": "Metadata transfer job resource schema for IoTSiteWise",
  "definitions": {
    "Name": {
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "Description": {
      "type": "string",
      "minLength": 1,
      "maxLength": 2048,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "ID": {
      "type": "string",
      "minLength": 36,
      "maxLength": 36,
      "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
    }
  }
}
```



```

    },
    "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",
      "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",
      "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"
    }
  }
}
}
```

Pantau data dengan alarm di AWS IoT SiteWise

Anda dapat mengonfigurasi alarm untuk data Anda untuk mengingatkan tim Anda saat peralatan atau proses berjalan secara sub-optimal. Kinerja optimal mesin atau proses berarti bahwa nilai untuk metrik tertentu harus berada dalam kisaran batas tinggi dan rendah. Ketika metrik ini berada di luar jangkauan operasinya, operator peralatan harus diberi tahu sehingga mereka dapat memperbaiki masalah. Gunakan alarm untuk mengidentifikasi masalah dengan cepat dan memberi tahu operator untuk memaksimalkan kinerja peralatan dan proses Anda.

Topik

- [Jenis alarm](#)
- [Status alarm](#)
- [Properti status alarm](#)
- [Tentukan alarm pada model aset di AWS IoT SiteWise](#)
- [Konfigurasi alarm pada aset di AWS IoT SiteWise](#)
- [Menanggapi alarm di AWS IoT SiteWise](#)
- [Menelan status alarm eksternal di AWS IoT SiteWise](#)

Jenis alarm

Anda dapat menentukan alarm yang mendeteksi di AWS Cloud dan alarm yang Anda deteksi dengan proses eksternal. AWS IoT SiteWise mendukung jenis alarm berikut:

- AWS IoT Events alarm

Note

Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS IoT Events. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses AWS IoT Events konsol atau AWS IoT Events sumber daya. Untuk informasi selengkapnya, lihat [AWS IoT Events akhir dukungan](#).

AWS IoT Events Alarm adalah alarm yang mendeteksi. AWS IoT Events AWS IoT SiteWise mengirimkan nilai properti aset ke model alarm di AWS IoT Events. Kemudian, AWS IoT Events

kirim status alarm ke AWS IoT SiteWise. Anda dapat mengonfigurasi opsi seperti saat alarm mendeteksi dan siapa yang harus diberi tahu saat status alarm berubah. Anda juga dapat menentukan [AWS IoT Events tindakan](#) yang terjadi ketika status alarm berubah.

Alarm AWS IoT Events dalam contoh model alarm. Model alarm menentukan ambang batas dan tingkat keparahan alarm, apa yang harus dilakukan ketika status alarm berubah, dan banyak lagi. Saat mengonfigurasi setiap sifat model alarm, Anda menentukan properti atribut dari model aset yang dipantau alarm. Semua aset berdasarkan model aset menggunakan nilai atribut saat AWS IoT Events mengevaluasi sifat alarm tersebut. Untuk informasi selengkapnya, lihat [Menggunakan alarm](#) di Panduan AWS IoT Events Pengembang.

Anda dapat merespons AWS IoT Events alarm saat berubah status. Misalnya, Anda dapat mengenali atau menunda alarm saat alarm menjadi aktif. Anda juga dapat mengaktifkan, menonaktifkan, dan mengatur ulang alarm.

SiteWise Pengguna monitor dapat memvisualisasikan, mengonfigurasi, dan merespons AWS IoT Events alarm di portal SiteWise Monitor. Untuk informasi selengkapnya, lihat [Memantau dengan alarm](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.

 Note

AWS IoT Events biaya berlaku untuk mengevaluasi alarm ini dan mentransfer data antara AWS IoT SiteWise dan AWS IoT Events. Untuk informasi selengkapnya, lihat [harga AWS IoT Events](#).

- Alarm eksternal

Alarm eksternal adalah alarm yang Anda evaluasi di luar. AWS IoT SiteWise Gunakan alarm eksternal jika Anda memiliki sumber data yang melaporkan status alarm. Alarm eksternal berisi properti pengukuran tempat Anda menelan data status alarm.

Anda tidak dapat mengakui atau menunda alarm eksternal saat berubah status.

SiteWise Pengguna monitor dapat melihat status alarm eksternal di portal SiteWise Monitor, tetapi mereka tidak dapat mengonfigurasi atau merespons alarm ini.

AWS IoT SiteWise tidak mengevaluasi keadaan alarm eksternal.

Status alarm

Alarm industri mencakup informasi tentang keadaan peralatan atau proses yang mereka pantau dan (opsional) informasi tentang respons operator terhadap status alarm.

Saat Anda menentukan AWS IoT Events alarm, Anda menentukan apakah akan mengaktifkan alur pengakuan atau tidak. Alur pengakuan diaktifkan secara default. Saat Anda mengaktifkan opsi ini, operator dapat mengenali alarm dan meninggalkan catatan dengan detail tentang alarm atau tindakan yang mereka ambil untuk mengatasinya. Jika operator tidak mengakui alarm aktif sebelum menjadi tidak aktif, alarm menjadi terkunci. Status terkunci menunjukkan bahwa alarm menjadi aktif dan tidak dikenali, sehingga operator perlu memeriksa peralatan atau proses dan mengakui alarm yang terkunci.

Alarm memiliki status berikut:

- **Normal (Normal)** - Alarm diaktifkan tetapi tidak aktif. Proses atau peralatan industri beroperasi seperti yang diharapkan.
- **Aktif (Active)** — Alarm aktif. Proses atau peralatan industri berada di luar jangkauan operasinya dan perlu diperhatikan.
- **Diakui (Acknowledged)** — Operator mengakui status alarm.

Status ini hanya berlaku untuk alarm tempat Anda mengaktifkan aliran pengakuan.

- **Latched (Latched)** — Alarm kembali normal tetapi aktif dan tidak ada operator yang mengakuinya. Proses atau peralatan industri membutuhkan perhatian dari operator untuk mengatur ulang alarm ke normal.

Status ini hanya berlaku untuk alarm tempat Anda mengaktifkan aliran pengakuan.

- **Snoozed (SnoozeDisabled)** — Alarm dinonaktifkan karena operator menunda alarm. Operator menentukan durasi alarm tunda. Setelah durasi itu, alarm kembali ke keadaan normal.
- **Disabled (Disabled)** - Alarm dinonaktifkan dan tidak akan mendeteksi.

Properti status alarm

AWS IoT SiteWise menyimpan data status alarm sebagai objek JSON yang diserialisasikan ke string. Objek ini berisi status dan informasi tambahan tentang alarm, seperti tindakan respons operator dan aturan yang dievaluasi alarm.

Anda mengidentifikasi properti status alarm dengan nama dan jenis strukturnya, `AWS/ALARM_STATE`. Untuk informasi selengkapnya, lihat [Tentukan alarm pada model aset di AWS IoT SiteWise](#).

Objek data status alarm berisi informasi berikut:

`stateName`

Status alarm. Untuk informasi selengkapnya, lihat [Status alarm](#).

Tipe data: `STRING`

`customerAction`

(Opsional) Objek yang berisi informasi tentang respons operator terhadap alarm. Operator dapat mengaktifkan, menonaktifkan, mengakui, dan menunda alarm. Ketika mereka melakukannya, data status alarm mencakup respons mereka dan catatan yang dapat mereka tinggalkan saat mereka merespons. Objek ini berisi informasi berikut:

`actionName`

Nama tindakan yang diambil operator untuk menanggapi alarm. Nilai ini berisi salah satu string berikut:

- `ENABLE`
- `DISABLE`
- `SNOOZE`
- `ACKNOWLEDGE`
- `RESET`

Tipe data: `STRING`

`enable`

(Opsional) Objek yang ada `customerAction` saat operator mengaktifkan alarm. Saat operator mengaktifkan alarm, status alarm berubah menjadi `Normal`. Objek ini berisi informasi berikut:

`note`

(Opsional) Catatan bahwa pelanggan pergi ketika mereka mengaktifkan alarm.

Tipe data: `STRING`

Panjang maksimal: 128 karakter

disable

(Opsional) Objek yang ada `customerAction` saat operator menonaktifkan alarm. Saat operator mengaktifkan alarm, status alarm berubah menjadi `Disabled`. Objek ini berisi informasi berikut:

note

(Opsional) Catatan bahwa pelanggan pergi ketika mereka menonaktifkan alarm.

Tipe data: `STRING`

Panjang maksimal: 128 karakter

acknowledge

(Opsional) Objek yang ada di `customerAction` saat operator mengakui alarm. Saat operator mengaktifkan alarm, status alarm berubah menjadi `Acknowledged`. Objek ini berisi informasi berikut:

note

(Opsional) Catatan bahwa pelanggan meninggalkan ketika mereka mengakui alarm.

Tipe data: `STRING`

Panjang maksimal: 128 karakter

snooze

(Opsional) Objek yang ada `customerAction` saat operator menunda alarm. Saat operator mengaktifkan alarm, status alarm berubah menjadi `SnoozeDisabled`. Objek ini berisi informasi berikut:

snoozeDuration

Durasi dalam hitungan detik operator menunda alarm. Alarm berubah menjadi `Normal` status setelah durasi ini.

Tipe data: `INTEGER`

note

(Opsional) Catatan bahwa pelanggan pergi ketika mereka menunda alarm.

Tipe data: `STRING`

Panjang maksimal: 128 karakter

ruleEvaluation

(Opsional) Objek yang berisi informasi tentang aturan yang mengevaluasi alarm. Objek ini berisi informasi berikut:

simpleRule

Objek yang berisi informasi tentang aturan sederhana, yang membandingkan nilai properti dengan nilai ambang batas dengan operator perbandingan. Objek ini berisi informasi berikut:

inputProperty

Nilai properti yang dievaluasi alarm ini.

Tipe data: DOUBLE

operator

Operator perbandingan yang digunakan alarm ini untuk membandingkan properti dengan ambang batas. Nilai ini berisi salah satu string berikut:

- <- Kurang dari
- <=- Kurang dari atau sama
- ==— Sama
- !=- Tidak sama
- >=Lebih besar dari atau sama
- >Lebih besar dari

Tipe data: STRING

threshold

Nilai ambang batas yang dibandingkan dengan nilai properti alarm ini.

Tipe data: DOUBLE

Tentukan alarm pada model aset di AWS IoT SiteWise

Model aset mendorong standardisasi data industri dan alarm Anda. Anda dapat menentukan definisi alarm pada model aset untuk membakukan alarm untuk semua aset berdasarkan model aset.

Anda menggunakan model aset komposit untuk menentukan alarm pada model aset. Model aset komposit adalah model aset yang menstandarisasi serangkaian properti tertentu pada model aset

lain. Model aset komposit memastikan bahwa properti tertentu hadir pada model aset. Alarm memiliki tipe, status, dan properti sumber (opsional), sehingga model komposit alarm memberlakukan bahwa properti ini ada.

Setiap model aset komposit memiliki tipe yang mendefinisikan properti untuk model komposit tersebut. Model komposit alarm menentukan properti untuk jenis alarm, status alarm, dan sumber alarm (opsional). Saat Anda membuat aset dari model aset dengan model komposit, aset tersebut menyertakan properti dari model komposit bersama properti yang Anda tentukan dalam model aset.

Setiap properti dalam model komposit harus memiliki nama yang mengidentifikasinya untuk jenis model komposit. Properti model komposit mendukung properti dengan tipe data yang kompleks. Properti ini memiliki tipe STRUCT data dan dataTypeSpec sifat yang menentukan tipe data kompleks properti. Properti tipe data yang kompleks berisi data JSON yang diserialkan sebagai string.

Model komposit alarm memiliki sifat berikut. Setiap properti harus memiliki nama yang mengidentifikasinya untuk jenis model komposit ini.

Jenis alarm

Jenis alarm. Tentukan satu dari yang berikut ini:

- **IOT_EVENTS**— AWS IoT Events Alarm. AWS IoT SiteWise mengirimkan data AWS IoT Events ke untuk mengevaluasi keadaan alarm ini. Anda harus menentukan properti sumber alarm untuk menentukan model AWS IoT Events alarm untuk definisi alarm ini.

Note

Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS IoT Events. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses AWS IoT Events konsol atau AWS IoT Events sumber daya. Untuk informasi selengkapnya, lihat [AWS IoT Events akhir dukungan](#).

- **EXTERNAL**— Alarm eksternal. Anda menelan keadaan alarm sebagai pengukuran.

Nama properti: `AWS/ALARM_TYPE`

Jenis properti: [atribut](#)

Tipe data: `STRING`

Status alarm

Data deret waktu untuk keadaan alarm. Ini adalah objek serial sebagai string yang berisi status dan informasi lain tentang alarm. Untuk informasi selengkapnya, lihat [Properti status alarm](#).

Nama properti: AWS/ALARM_STATE

Jenis properti: [pengukuran](#)

Tipe data: STRUCT

Jenis struktur data: AWS/ALARM_STATE

Sumber alarm

(Opsional) Nama Sumber Daya Amazon (ARN) dari sumber daya yang mengevaluasi status alarm. Untuk AWS IoT Events alarm, ini adalah ARN dari model alarm.

Nama properti: AWS/ALARM_SOURCE

Jenis properti: [atribut](#)

Tipe data: STRING

Example Contoh model komposit alarm

Model aset berikut mewakili boiler yang memiliki alarm untuk memantau suhunya. AWS IoT SiteWise mengirimkan data suhu AWS IoT Events untuk mendeteksi alarm.

```
{
  "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": {}
        }
      }
    ]
  }
]
}

```

Topik

- [Persyaratan untuk pemberitahuan alarm di AWS IoT SiteWise](#)
- [Tentukan AWS IoT Events alarm untuk AWS IoT SiteWise](#)
- [Tentukan alarm eksternal di AWS IoT SiteWise](#)

Persyaratan untuk pemberitahuan alarm di AWS IoT SiteWise

AWS IoT Events menggunakan AWS Lambda fungsi di AWS akun Anda untuk mengirim pemberitahuan alarm. Anda harus membuat fungsi Lambda ini di AWS Wilayah yang sama dengan alarm Anda untuk mengaktifkan pemberitahuan alarm. Fungsi Lambda ini menggunakan [Amazon Simple Notification Service \(Amazon SNS\)](#) untuk mengirim notifikasi teks dan [Amazon Simple Email Service \(Amazon SES\)](#) untuk mengirim pemberitahuan email. Saat Anda membuat AWS IoT Events alarm, Anda mengonfigurasi protokol dan pengaturan yang digunakan alarm untuk mengirim pemberitahuan.

Note

Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS IoT Events. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses AWS IoT Events konsol atau AWS IoT Events sumber daya. Untuk informasi selengkapnya, lihat [AWS IoT Events akhir dukungan](#).

AWS IoT Events menyediakan template AWS CloudFormation tumpukan yang dapat Anda gunakan untuk membuat fungsi Lambda ini di akun Anda. Untuk informasi selengkapnya, lihat [Fungsi Lambda pemberitahuan alarm](#) di Panduan AWS IoT Events Pengembang.

Tentukan AWS IoT Events alarm untuk AWS IoT SiteWise

Note

Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS IoT Events. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses AWS IoT Events konsol atau AWS IoT Events sumber daya. Untuk informasi selengkapnya, lihat [AWS IoT Events akhir dukungan](#).

Saat Anda membuat AWS IoT Events alarm, AWS IoT SiteWise kirimkan nilai properti aset AWS IoT Events ke untuk mengevaluasi status alarm. AWS IoT Events definisi alarm bergantung pada model alarm yang Anda tentukan AWS IoT Events. Untuk menentukan AWS IoT Events alarm pada model aset, Anda menentukan model komposit alarm yang menentukan model AWS IoT Events alarm sebagai properti sumber alarmnya.

AWS IoT Events alarm tergantung pada input seperti ambang alarm dan pengaturan pemberitahuan alarm. Anda mendefinisikan input ini sebagai atribut pada model aset. Anda kemudian dapat menyesuaikan input ini pada setiap aset berdasarkan model. AWS IoT SiteWise Konsol dapat membuat atribut ini untuk Anda. Jika Anda mendefinisikan alarm dengan API AWS CLI atau, Anda harus menentukan atribut ini secara manual pada model aset.

Anda juga dapat menentukan tindakan lain yang terjadi ketika alarm Anda mendeteksi, seperti tindakan pemberitahuan alarm khusus. Misalnya, Anda dapat mengonfigurasi tindakan yang mengirimkan pemberitahuan push ke topik Amazon SNS. Untuk informasi selengkapnya tentang tindakan yang dapat Anda tentukan, lihat [Bekerja dengan AWS layanan lain](#) di Panduan AWS IoT Events Pengembang.

Saat Anda memperbarui atau menghapus model aset, AWS IoT SiteWise dapat memeriksa apakah model alarm di AWS IoT Events memantau properti aset yang terkait dengan model aset ini. Ini mencegah Anda menghapus properti aset yang saat ini digunakan AWS IoT Events alarm. Untuk mengaktifkan fitur ini AWS IoT SiteWise, Anda harus memiliki `iotevents:ListInputRoutings` izin. Izin ini memungkinkan AWS IoT SiteWise untuk melakukan panggilan ke operasi [ListInputRoutings](#) API yang didukung oleh AWS IoT Events. Untuk informasi selengkapnya, lihat [ListInputRoutings Izin \(Opsional\)](#).

 Note

Fitur pemberitahuan alarm tidak tersedia di Wilayah Tiongkok (Beijing).

Topik

- [Tentukan AWS IoT Events alarm \(AWS IoT SiteWise konsol\)](#)
- [Tentukan AWS IoT Events alarm \(AWS IoT Events konsol\)](#)
- [Tentukan AWS IoT Events alarm \(AWS CLI\)](#)

Tentukan AWS IoT Events alarm (AWS IoT SiteWise konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menentukan AWS IoT Events alarm pada model aset yang ada. Untuk menentukan AWS IoT Events alarm pada model aset baru, buat model aset, lalu selesaikan langkah-langkah ini. Untuk informasi selengkapnya, lihat [Buat model aset di AWS IoT SiteWise](#).

Important

Setiap alarm memerlukan atribut yang menentukan nilai ambang untuk dibandingkan dengan alarm. Anda harus menentukan atribut nilai ambang pada model aset sebelum Anda dapat menentukan alarm.

Pertimbangkan contoh di mana Anda ingin menentukan alarm yang mendeteksi ketika turbin angin melebihi peringkat kecepatan angin maksimum 50 mph. Sebelum Anda menentukan alarm, Anda harus menentukan atribut (Kecepatan angin maksimum) dengan nilai default 50.

Untuk menentukan AWS IoT Events alarm pada model aset

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset untuk menentukan alarm.
4. Pilih tab Alarm.
5. Pilih Tambahkan alarm.
6. Di bagian Opsi jenis alarm, pilih AWS IoT Events alarm.
7. Di bagian Detail alarm, lakukan hal berikut:
 - a. Masukkan nama untuk alarm Anda.
 - b. (Opsional) Masukkan deskripsi untuk alarm Anda.
8. Di bagian Definisi ambang batas, Anda menentukan kapan alarm mendeteksi dan tingkat keparahan alarm. Lakukan hal-hal berikut:
 - a. Pilih Properti tempat alarm mendeteksi. Setiap kali properti ini menerima nilai baru, AWS IoT SiteWise mengirimkan nilai AWS IoT Events ke untuk mengevaluasi keadaan alarm.
 - b. Pilih Operator yang akan digunakan untuk membandingkan properti dengan nilai ambang batas. Pilih dari salah satu pilihan berikut:

- < kurang dari
 - <= kurang dari atau sama
 - == sama
 - != tidak sama
 - >= lebih besar dari atau sama
 - > lebih besar dari
- c. Untuk Nilai, pilih properti atribut yang akan digunakan sebagai nilai ambang batas. AWS IoT Events membandingkan nilai properti dengan nilai atribut ini.
- d. Masukkan tingkat keparahan alarm. Gunakan nomor yang dipahami tim Anda untuk mencerminkan tingkat keparahan alarm ini.
9. (Opsional) Di pengaturan Pemberitahuan - bagian opsional, lakukan hal berikut:
- a. Pilih Aktif.

 Note

Jika Anda memilih Tidak Aktif, Anda dan tim Anda tidak akan menerima pemberitahuan alarm apa pun.

- b. Untuk Penerima, pilih penerima.

 Important

Anda dapat mengirim pemberitahuan alarm ke AWS IAM Identity Center pengguna. Untuk menggunakan fitur ini, Anda harus mengaktifkan IAM Identity Center. Anda hanya dapat mengaktifkan Pusat Identitas IAM di satu AWS Wilayah pada satu waktu. Ini berarti Anda dapat menentukan pemberitahuan alarm hanya di Wilayah tempat Anda mengaktifkan Pusat Identitas IAM. Untuk informasi selengkapnya, lihat [Memulai](#) di Panduan AWS IAM Identity Center Pengguna.

- c. Untuk Protokol, pilih dari opsi berikut:
- Email & teks - Alarm memberi tahu pengguna IAM Identity Center dengan pesan SMS dan pesan email.
 - Email — Alarm memberi tahu pengguna IAM Identity Center dengan pesan email.

- Teks — Alarm memberi tahu pengguna IAM Identity Center dengan pesan SMS.

d. Untuk Pengirim, pilih pengirim.

 Important

Anda harus memverifikasi alamat email pengirim di Amazon Simple Email Service (Amazon SES). Untuk informasi selengkapnya, lihat [Memverifikasi identitas alamat email](#), di Panduan Pengembang Layanan Email Sederhana Amazon.

10. Di bagian Status aset default, Anda dapat mengatur status default untuk alarm yang dibuat dari model aset ini.

 Note

Anda mengaktifkan atau menonaktifkan alarm ini untuk aset yang Anda buat dari model aset ini di langkah selanjutnya.

11. Di bagian Pengaturan lanjutan, Anda dapat mengonfigurasi izin, pengaturan pemberitahuan tambahan, tindakan status alarm, model alarm di SiteWise Monitor, dan aliran pengakuan.

 Note

AWS IoT Events alarm memerlukan peran layanan berikut:

- Peran yang AWS IoT Events mengasumsikan untuk mengirim nilai status alarm ke AWS IoT SiteWise.
- Peran yang AWS IoT Events mengasumsikan untuk mengirim data ke Lambda. Anda hanya perlu peran ini jika alarm Anda mengirimkan notifikasi.

Di bagian Izin, lakukan hal berikut:

- a. Untuk AWS IoT Events peran, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `iotsitewise:BatchPutAssetPropertyValue` izin dan hubungan kepercayaan yang memungkinkan `iotevents.amazonaws.com` untuk mengambil peran tersebut.
- b. Untuk peran AWS IoT Events Lambda, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `lambda:InvokeFunction` dan `ss-`

`directory:DescribeUser` izin dan hubungan kepercayaan yang memungkinkan `iotevents.amazonaws.com` untuk mengambil peran.

12. (Opsional) Di bagian Pengaturan pemberitahuan tambahan, lakukan hal berikut:

- a. Untuk atribut Penerima, Anda menentukan atribut yang nilainya menentukan penerima pemberitahuan. Anda dapat memilih pengguna IAM Identity Center sebagai penerima.

Anda dapat membuat atribut atau menggunakan atribut yang ada pada model aset.

- Jika Anda memilih Buat atribut penerima baru, tentukan nama atribut Penerima dan nilai default Penerima - opsional untuk atribut.
- Jika Anda memilih Gunakan atribut penerima yang ada, pilih atribut di Nama atribut Penerima. Alarm menggunakan nilai default atribut yang Anda pilih.

Anda dapat mengganti nilai default pada setiap aset yang Anda buat dari model aset ini.

- b. Untuk atribut pesan kustom, Anda menentukan atribut yang nilainya menentukan pesan kustom yang akan dikirim selain pesan perubahan status default. Misalnya, Anda dapat menentukan pesan yang membantu tim Anda memahami cara mengatasi alarm ini.

Anda dapat memilih untuk membuat atribut atau menggunakan atribut yang ada pada model aset.

- Jika Anda memilih untuk Membuat atribut pesan kustom baru, tentukan nama atribut pesan kustom dan Nilai default pesan kustom - opsional untuk atribut.
- Jika Anda memilih Gunakan atribut pesan kustom yang ada, pilih atribut di Nama atribut pesan kustom. Alarm menggunakan nilai default atribut yang Anda pilih.

Anda dapat mengganti nilai default pada setiap aset yang Anda buat dari model aset ini.

- c. Untuk Kelola fungsi Lambda Anda, lakukan salah satu hal berikut:
 - Untuk AWS IoT SiteWise membuat fungsi Lambda baru, pilih Buat lambda baru dari templat yang dikelola AWS.
 - Untuk menggunakan fungsi Lambda yang ada, pilih Gunakan lambda yang ada dan pilih nama fungsi.

Untuk informasi selengkapnya, lihat [Mengelola notifikasi alarm](#) di Panduan AWS IoT Events Pengembang.

13. (Opsional) Di bagian Set state action, lakukan hal berikut:

- a. Pilih Edit tindakan.
- b. Di bawah Tambahkan tindakan status alarm, tambahkan tindakan. dan pilih Simpan.

Anda dapat menambahkan hingga 10 tindakan.

AWS IoT Events dapat melakukan tindakan saat alarm aktif. Anda dapat menentukan tindakan bawaan untuk menggunakan timer atau menyetel variabel, atau mengirim data ke AWS sumber daya lain. Untuk informasi selengkapnya, lihat [Tindakan yang didukung](#) di Panduan AWS IoT Events Pengembang.

14. (Opsional) Di bawah Kelola model alarm di SiteWise Monitor - opsional, pilih Aktif atau Tidak Aktif.

Gunakan opsi ini sehingga Anda dapat memperbarui model alarm di SiteWise Monitorss. Pengaturan ini diaktifkan secara default.

15. Di bawah Akui aliran, pilih Aktif atau Tidak Aktif. Untuk informasi lebih lanjut tentang alur pengakuan, lihat [Status alarm](#).

16. Pilih Tambahkan alarm.

 Note

AWS IoT SiteWise Konsol membuat beberapa permintaan API untuk menambahkan alarm ke model aset. Saat Anda memilih Tambah alarm, konsol akan membuka kotak dialog yang menunjukkan kemajuan permintaan API ini. Tetap di halaman ini sampai setiap permintaan API berhasil atau sampai permintaan API gagal. Jika permintaan gagal, tutup kotak dialog, perbaiki masalah, dan pilih Tambahkan alarm untuk mencoba lagi.

Tentukan AWS IoT Events alarm (AWS IoT Events konsol)

Anda dapat menggunakan AWS IoT Events konsol untuk menentukan AWS IoT Events alarm pada model aset yang ada. Untuk menentukan AWS IoT Events alarm pada model aset baru, buat model aset, lalu selesaikan langkah-langkah ini. Untuk informasi selengkapnya, lihat [Buat model aset di AWS IoT SiteWise](#).

Important

Setiap alarm memerlukan atribut yang menentukan nilai ambang untuk dibandingkan dengan alarm. Anda harus menentukan atribut nilai ambang pada model aset sebelum Anda dapat menentukan alarm.

Pertimbangkan contoh di mana Anda ingin menentukan alarm yang mendeteksi ketika turbin angin melebihi peringkat kecepatan angin maksimum 50 mph. Sebelum Anda menentukan alarm, Anda harus menentukan atribut (Kecepatan angin maksimum) dengan nilai default 50.

Untuk menentukan AWS IoT Events alarm pada model aset

1. Navigasikan ke [konsol AWS IoT Events](#) tersebut.
2. Di panel navigasi, pilih Model alarm.
3. Pilih Buat model alarm.
4. Masukkan nama untuk alarm Anda.
5. (Opsional) Masukkan deskripsi untuk alarm Anda.
6. Di bagian Target alarm, lakukan hal berikut:
 - a. Untuk opsi Target, pilih properti AWS IoT SiteWise aset.
 - b. Pilih model aset yang ingin Anda tambahkan alarmnya.
7. Di bagian Definisi ambang batas, Anda menentukan kapan alarm mendeteksi dan tingkat keparahan alarm. Lakukan hal-hal berikut:
 - a. Pilih Properti tempat alarm mendeteksi. Setiap kali properti ini menerima nilai baru, AWS IoT SiteWise mengirimkan nilai AWS IoT Events ke untuk mengevaluasi keadaan alarm.
 - b. Pilih Operator yang akan digunakan untuk membandingkan properti dengan nilai ambang batas. Pilih dari salah satu pilihan berikut:
 - < kurang dari

- <= kurang dari atau sama
 - == sama
 - != tidak sama
 - >= lebih besar dari atau sama
 - > lebih besar dari
- c. Untuk Nilai, pilih properti atribut yang akan digunakan sebagai nilai ambang batas. AWS IoT Events membandingkan nilai properti dengan nilai atribut ini.
- d. Masukkan tingkat keparahan alarm. Gunakan nomor yang dipahami tim Anda untuk mencerminkan tingkat keparahan alarm ini.
8. (Opsional) Di pengaturan Pemberitahuan - bagian opsional, lakukan hal berikut:
- a. Untuk Protokol, pilih dari opsi berikut:
- Email & teks - Alarm memberi tahu pengguna IAM Identity Center dengan pesan SMS dan pesan email.
 - Email — Alarm memberi tahu pengguna IAM Identity Center dengan pesan email.
 - Teks — Alarm memberi tahu pengguna IAM Identity Center dengan pesan SMS.
- b. Untuk Pengirim, pilih pengirim.
-  **Important**

Anda harus memverifikasi alamat email pengirim di Amazon Simple Email Service (Amazon SES). Untuk informasi selengkapnya, lihat [Memverifikasi alamat email di Amazon SES](#), di Panduan Pengembang Layanan Email Sederhana Amazon.
- c. Pilih atribut di atribut Penerima - opsional. Alarm menggunakan nilai default atribut yang Anda pilih.
- d. Pilih atribut di atribut pesan kustom - opsional. Alarm menggunakan nilai default atribut yang Anda pilih.
9. Di bagian Instance, tentukan status Default untuk alarm ini. Anda dapat mengaktifkan atau menonaktifkan alarm ini untuk semua aset yang Anda buat dari model aset ini di langkah selanjutnya.
10. Dalam pengaturan Pengaturan lanjutan, Anda dapat mengonfigurasi izin, pengaturan pemberitahuan tambahan, tindakan status alarm, model alarm di SiteWise Monitor, dan aliran pengakuan.

 Note

AWS IoT Events alarm memerlukan peran layanan berikut:

- Peran yang AWS IoT Events mengasumsikan untuk mengirim nilai status alarm ke AWS IoT SiteWise.
- Peran yang AWS IoT Events mengasumsikan untuk mengirim data ke Lambda. Anda hanya perlu peran ini jika alarm Anda mengirimkan notifikasi.

- a. Di bagian Akui alur, pilih Diaktifkan atau Dinonaktifkan. Untuk informasi lebih lanjut tentang alur pengakuan, lihat [Status alarm](#).
- b. Di bagian Izin, lakukan hal berikut:
 - i. Untuk AWS IoT Events peran, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `iotsitewise:BatchPutAssetPropertyValue` izin dan hubungan kepercayaan yang memungkinkan `iotevents.amazonaws.com` untuk mengambil peran tersebut.
 - ii. Untuk peran Lambda, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `lambda:InvokeFunction` dan `sso-directory:DescribeUser` izin dan hubungan kepercayaan yang memungkinkan `iotevents.amazonaws.com` untuk mengambil peran.
- c. (Opsional) Di panel Pengaturan pemberitahuan tambahan, lakukan hal berikut:
 - Untuk Kelola fungsi Lambda Anda, lakukan salah satu hal berikut:
 - Untuk AWS IoT Events membuat fungsi Lambda baru, pilih Buat fungsi Lambda baru.
 - Untuk menggunakan fungsi Lambda yang ada, pilih Gunakan fungsi Lambda yang ada dan pilih nama fungsi.

Untuk informasi selengkapnya, lihat [Mengelola notifikasi alarm](#) di Panduan AWS IoT Events Pengembang.

- d. (Opsional) Di bagian Set state action - opsional, lakukan hal berikut:
 - Di bawah Tindakan status alarm, tambahkan tindakan. dan pilih Simpan.

Anda dapat menambahkan hingga 10 tindakan.

AWS IoT Events dapat melakukan tindakan saat alarm aktif. Anda dapat menentukan tindakan bawaan untuk menggunakan timer atau menyetel variabel, atau mengirim data ke AWS sumber daya lain. Untuk informasi selengkapnya, lihat [Tindakan yang didukung](#) di Panduan AWS IoT Events Pengembang.

11. Pilih Buat.

Note

AWS IoT Events Konsol membuat beberapa permintaan API untuk menambahkan alarm ke model aset. Saat Anda memilih Tambah alarm, konsol akan membuka kotak dialog yang menunjukkan kemajuan permintaan API ini. Tetap di halaman ini sampai setiap permintaan API berhasil atau sampai permintaan API gagal. Jika permintaan gagal, tutup kotak dialog, perbaiki masalah, dan pilih Tambahkan alarm untuk mencoba lagi.

Tentukan AWS IoT Events alarm (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menentukan AWS IoT Events alarm yang memantau properti aset. Anda dapat menentukan alarm pada model aset baru atau yang sudah ada. Setelah Anda menentukan alarm pada model aset, Anda membuat alarm AWS IoT Events dan menghubungkannya ke model aset. Dalam proses ini, Anda melakukan hal berikut:

Langkah-langkah

- [Langkah 1: Tentukan alarm pada model aset](#)
- [Langkah 2: Tentukan model AWS IoT Events alarm](#)
- [Langkah 3: Aktifkan aliran data antara AWS IoT SiteWise dan AWS IoT Events](#)

Langkah 1: Tentukan alarm pada model aset

Tambahkan definisi alarm dan properti terkait ke model aset baru atau yang sudah ada.

Untuk menentukan alarm pada model aset (CLI)

1. Buat file bernama `asset-model-payload.json`. Ikuti langkah-langkah di bagian lain ini untuk menambahkan detail model aset Anda ke file, tetapi jangan kirimkan permintaan untuk membuat atau memperbarui model aset. Di bagian ini, Anda menambahkan definisi alarm ke detail model aset dalam `asset-model-payload.json` file.
 - Untuk informasi selengkapnya tentang cara membuat model aset, lihat [Buat model aset \(AWS CLI\)](#).
 - Untuk informasi selengkapnya tentang cara memperbarui model aset yang ada, lihat [Memperbarui model aset atau komponen \(AWS CLI\)](#).

Note

Model aset Anda harus menentukan setidaknya satu properti aset, termasuk properti aset untuk dipantau dengan alarm.

2. Tambahkan model komposit alarm (`assetModelCompositeModels`) ke model aset. Model komposit AWS IoT Events alarm menentukan `IOT_EVENTS` jenis dan menentukan properti sumber alarm. Anda menambahkan properti sumber alarm setelah Anda membuat model alarm di AWS IoT Events.

Important

Model komposit alarm harus memiliki nama yang sama dengan model AWS IoT Events alarm yang Anda buat nanti. Nama model alarm hanya dapat berisi karakter alfanumerik. Tentukan nama alfanumerik yang unik sehingga Anda dapat menggunakan nama yang sama untuk model alarm.

```
{
  ...
  "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. Tambahkan atribut ambang alarm ke model aset. Tentukan nilai default yang akan digunakan untuk ambang batas ini. Anda dapat mengganti nilai default ini pada setiap aset berdasarkan model ini.

Note

Atribut ambang alarm harus berupa INTEGER atau aDOUBLE.

```

{
    ...
    "assetModelProperties": [
        ...
        {
            "name": "Temperature Max Threshold",
            "dataType": "DOUBLE",
            "type": {
                "attribute": {
                    "defaultValue": "105.0"
                }
            }
        }
    ]
}

```



```
]
}
```

4. (Opsional) Tambahkan atribut notifikasi alarm ke model aset. Atribut ini menentukan penerima IAM Identity Center dan input lain yang AWS IoT Events digunakan untuk mengirim notifikasi saat alarm berubah status. Anda dapat mengganti default ini pada setiap aset berdasarkan model ini.

Important

Anda dapat mengirim pemberitahuan alarm ke AWS IAM Identity Center pengguna. Untuk menggunakan fitur ini, Anda harus mengaktifkan IAM Identity Center. Anda hanya dapat mengaktifkan Pusat Identitas IAM di satu AWS Wilayah pada satu waktu. Ini berarti Anda dapat menentukan pemberitahuan alarm hanya di Wilayah tempat Anda mengaktifkan Pusat Identitas IAM. Untuk informasi selengkapnya, lihat [Memulai](#) di Panduan AWS IAM Identity Center Pengguna.

Lakukan hal-hal berikut:

- a. Tambahkan atribut yang menentukan ID toko identitas Pusat Identitas IAM Anda. Anda dapat menggunakan operasi [ListInstances](#) API Pusat Identitas IAM untuk membuat daftar toko identitas Anda. Operasi ini hanya berfungsi di Wilayah tempat Anda mengaktifkan Pusat Identitas IAM.

```
aws sso-admin list-instances
```

Kemudian, tentukan ID penyimpanan identitas (misalnya, d-123EXAMPLE) sebagai nilai default untuk atribut.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
```

```

        "defaultValue": "d-123EXAMPLE"
      }
    }
  }
]
}

```

- b. Tambahkan atribut yang menentukan ID pengguna Pusat Identitas IAM yang menerima pemberitahuan. Untuk menentukan penerima notifikasi default, tambahkan ID pengguna IAM Identity Center sebagai nilai default. Lakukan salah satu hal berikut untuk mendapatkan ID pengguna IAM Identity Center:
 - i. Anda dapat menggunakan IAM Identity Center [ListUsers](#) API untuk mendapatkan ID pengguna yang nama penggunanya Anda kenal. Ganti *d-123EXAMPLE* dengan ID toko identitas Anda, dan ganti *Name* dengan nama pengguna pengguna.

```

aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
  --filters AttributePath=UserName,AttributeValue=Name

```

- ii. Gunakan [konsol Pusat Identitas IAM](#) untuk menelusuri pengguna Anda dan menemukan ID pengguna.

Kemudian, tentukan ID pengguna (misalnya, 123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE) sebagai nilai default untuk atribut, atau tentukan atribut tanpa nilai default.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
      }
    }
  ]
}

```

```
}
```

- c. (Opsional) Tambahkan atribut yang menentukan ID pengirim default untuk pemberitahuan pesan SMS (teks). ID pengirim ditampilkan sebagai pengirim pesan pada pesan yang dikirim oleh Amazon Simple Notification Service (Amazon SNS). Untuk informasi selengkapnya, lihat [Meminta ID pengirim AWS Olah Pesan Pengguna Akhir SMS di Panduan AWS Olah Pesan Pengguna Akhir SMS Pengguna](#).

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}
```

- d. (Opsional) Tambahkan atribut yang menentukan alamat email default untuk digunakan sebagai alamat from dalam pemberitahuan email.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "fromAddress",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "my.factory@example.com"
        }
      }
    }
  ]
}
```

- e. (Opsional) Tambahkan atribut yang menentukan subjek default untuk digunakan dalam pemberitahuan email.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "[ALERT] High boiler temperature"
        }
      }
    }
  ]
}
```

- f. (Opsional) Tambahkan atribut yang menentukan pesan tambahan untuk disertakan dalam pemberitahuan. Secara default, pesan notifikasi menyertakan informasi tentang alarm. Anda juga dapat menyertakan pesan tambahan yang memberi pengguna lebih banyak informasi..

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Turn off the power before you check the alarm."
        }
      }
    }
  ]
}
```

5. Buat model aset atau perbarui model aset yang ada. Lakukan salah satu tindakan berikut:
- Untuk membuat model aset, jalankan perintah berikut.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Untuk memperbarui model aset yang ada, jalankan perintah berikut. Ganti *asset-model-id* dengan ID model aset.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Setelah Anda menjalankan perintah, perhatikan `assetModelId` dalam respon.

Contoh: Model aset boiler

Model aset berikut mewakili boiler yang melaporkan data suhu. Model aset ini mendefinisikan alarm yang mendeteksi ketika boiler terlalu panas.

```
{
  "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": {}
          }
        }
      ]
    }
  ]
}

```

Langkah 2: Tentukan model AWS IoT Events alarm

Buat model alarm di AWS IoT Events. Di AWS IoT Events, Anda menggunakan ekspresi untuk menentukan nilai dalam model alarm. Anda dapat menggunakan ekspresi untuk menentukan nilai dari AWS IoT SiteWise untuk mengevaluasi dan digunakan sebagai input ke alarm. Saat AWS IoT

SiteWise mengirim nilai properti aset ke model alarm, AWS IoT Events mengevaluasi ekspresi untuk mendapatkan nilai properti atau ID aset. Anda dapat menggunakan ekspresi berikut dalam model alarm:

- Nilai properti aset

Untuk mendapatkan nilai properti aset, gunakan ekspresi berikut. Ganti *assetModelId* dengan ID model aset dan ganti *propertyId* dengan ID properti.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.propertyValue.value
```

- Aset IDs

Untuk mendapatkan ID aset, gunakan ekspresi berikut. Ganti *assetModelId* dengan ID model aset dan ganti *propertyId* dengan ID properti.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

Note

Saat Anda membuat model alarm, Anda dapat menentukan literal alih-alih ekspresi yang mengevaluasi AWS IoT SiteWise nilai. Ini dapat mengurangi jumlah atribut yang Anda tentukan pada model aset Anda. Namun, jika Anda mendefinisikan nilai sebagai literal, Anda tidak dapat menyesuaikan nilai tersebut pada aset berdasarkan model aset. AWS IoT SiteWise Monitor Pengguna Anda juga tidak dapat menyesuaikan alarm, karena mereka dapat mengonfigurasi pengaturan alarm hanya pada aset.

Untuk membuat model AWS IoT Events alarm (CLI)

1. Saat Anda membuat model alarm AWS IoT Events, Anda harus menentukan ID dari setiap properti yang digunakan alarm, yang mencakup yang berikut ini:
 - Properti status alarm dalam model aset komposit
 - Properti yang dipantau alarm
 - Atribut ambang
 - (Opsional) Atribut ID penyimpanan identitas Pusat Identitas IAM

- (Opsional) Atribut ID pengguna IAM Identity Center
- (Opsional) Atribut ID pengirim SMS
- (Opsional) Atribut email dari alamat
- (Opsional) Atribut subjek email
- (Opsional) Atribut pesan tambahan

Jalankan perintah berikut untuk mengambil IDs properti ini pada model aset. Ganti *asset-model-id* dengan ID model aset dari langkah sebelumnya.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Operasi mengembalikan respons yang berisi detail model aset. Perhatikan ID setiap properti yang digunakan alarm. Anda menggunakan ini IDs ketika Anda membuat model AWS IoT Events alarm di langkah berikutnya.

2. Buat model alarm di AWS IoT Events. Lakukan hal-hal berikut:
 - a. Buat file bernama `alarm-model-payload.json`.
 - b. Salin objek JSON berikut ke dalam file.
 - c. Masukkan nama (`alarmModelName`), deskripsi (`alarmModelDescription`), dan tingkat keparahan (`severity`) untuk alarm Anda. Untuk tingkat keparahan, tentukan bilangan bulat yang mencerminkan tingkat keparahan perusahaan Anda.

 Important

Model alarm harus memiliki nama yang sama dengan model komposit alarm yang Anda tentukan pada model aset Anda sebelumnya.
Nama model alarm hanya dapat berisi karakter alfanumerik.

```
{  
  "alarmModelName": "BoilerTemperatureHighAlarm",  
  "alarmModelDescription": "Detects when the boiler temperature is high.",  
  "severity": 3  
}
```

- d. Tambahkan aturan perbandingan (`alarmRule`) ke alarm. Aturan ini mendefinisikan properti untuk monitor (`inputProperty`), nilai ambang untuk membandingkan (`threshold`), dan operator perbandingan untuk menggunakan (`comparisonOperator`).
- Ganti `assetModelId` dengan ID model aset.
 - Ganti `alarmPropertyId` dengan ID properti yang dipantau alarm.
 - Ganti `thresholdAttributeId` dengan ID properti atribut threshold.
 - Ganti `GREATER` dengan operator yang akan digunakan untuk membandingkan nilai properti dengan ambang batas. Pilih dari salah satu pilihan berikut:
 - 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. Tambahkan action (`alarmEventActions`) untuk mengirim status alarm ke AWS IoT SiteWise saat alarm berubah status.

Note

Untuk konfigurasi lanjutan, Anda dapat menentukan tindakan tambahan yang harus dilakukan saat alarm berubah status. Misalnya, Anda mungkin memanggil AWS Lambda fungsi atau memublikasikan ke topik MQTT. Untuk informasi selengkapnya, lihat [Bekerja dengan AWS layanan lain](#) di Panduan AWS IoT Events Pengembang.

- Ganti *assetModelId* dengan ID model aset.
- Ganti *alarmPropertyId* dengan ID properti yang dipantau alarm.
- Ganti *alarmStatePropertyId* dengan ID properti status alarm dalam model komposit alarm.

```
{
  "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. (Opsional) Konfigurasi pengaturan pemberitahuan alarm. Tindakan pemberitahuan alarm menggunakan fungsi Lambda di akun Anda untuk mengirim pemberitahuan alarm. Untuk informasi selengkapnya, lihat [Persyaratan untuk pemberitahuan alarm di AWS IoT SiteWise](#). Dalam pengaturan pemberitahuan alarm, Anda dapat mengonfigurasi notifikasi SMS dan email untuk dikirim ke pengguna IAM Identity Center. Lakukan hal-hal berikut:
 - i. Tambahkan konfigurasi notifikasi alarm (`alarmNotification`) ke payload `alarm-model-payload.json`.
 - Ganti `alarmNotificationFunctionArn` dengan ARN dari fungsi Lambda yang menangani notifikasi alarm.

```
{
  "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. (Opsional) Konfigurasi notifikasi SMS (`smsConfigurations`) untuk dikirim ke pengguna Pusat Identitas IAM saat alarm berubah status.
 - Ganti *identityStoreIdAttributeId* dengan ID atribut yang berisi ID penyimpanan identitas Pusat Identitas IAM.
 - Ganti *userIdAttributeId* dengan ID atribut yang berisi ID pengguna IAM Identity Center.
 - Ganti *senderIdAttributeId* dengan ID atribut yang berisi ID pengirim Amazon SNS, atau hapus `senderId` dari payload.
 - Ganti *additionalMessageAttributeId* dengan ID atribut yang berisi pesan tambahan, atau hapus `additionalMessage` dari payload.

```

{
  "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":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.va
                "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
              }
            }
          ],
          "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId` .propertyValue.value",
          "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.
        }
      ]
    }
  ]
}
}
}
}

```

iii. (Opsional) Konfigurasi notifikasi email (`emailConfigurations`) untuk dikirim ke pengguna Pusat Identitas IAM saat alarm berubah status.

- Ganti *identityStoreIdAttributeId* dengan ID properti atribut ID penyimpanan identitas Pusat Identitas IAM.
- Ganti *userIdAttributeId* dengan ID properti atribut ID pengguna Pusat Identitas IAM.

- Ganti *fromAddressAttributeId* dengan ID properti atribut alamat “dari”, atau hapus from dari muatan.
- Ganti *emailSubjectAttributeId* dengan ID properti atribut subjek email, atau hapus subject dari payload.
- Ganti *additionalMessageAttributeId* dengan ID properti atribut pesan tambahan, atau hapus additionalMessage dari payload.

```
{
  "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"
            }
        }
    ]
}

```


}

- g. (Opsional) Tambahkan kemampuan alarm (`alarmCapabilities`) ke payload `dialarm-model-payload.json`. Dalam objek ini, Anda dapat menentukan apakah alur pengakuan diaktifkan dan status aktifkan default untuk aset berdasarkan model aset. Untuk informasi lebih lanjut tentang alur pengakuan, lihat [Status alarm](#).

```
{
  "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": true
    }
  }
}

```

- h. Tambahkan peran layanan IAM (`roleArn`) yang AWS IoT Events dapat mengasumsikan untuk mengirim data ke AWS IoT SiteWise. Peran ini membutuhkan `iotsitewise:BatchPutAssetPropertyValue` izin dan hubungan kepercayaan yang memungkinkan `iotevents.amazonaws.com` untuk mengambil peran. Untuk mengirim notifikasi, peran ini juga memerlukan `sso-directory:DescribeUser` izin `lambda:InvokeFunction` dan. Untuk informasi selengkapnya, lihat [Peran layanan alarm](#) di Panduan AWS IoT Events Pengembang.
- Ganti `roleArn` dengan ARN dari peran yang AWS IoT Events dapat diasumsikan untuk melakukan tindakan ini.

```

{
  "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. Jalankan perintah berikut untuk membuat model AWS IoT Events alarm dari payload `dialarm-model-payload.json`.

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. Operasi mengembalikan respons yang mencakup ARN dari model alarm, `alarmModelArn`. Salin ARN ini untuk mengatur definisi alarm pada model aset Anda di langkah berikutnya.

Langkah 3: Aktifkan aliran data antara AWS IoT SiteWise dan AWS IoT Events

Setelah Anda membuat sumber daya yang diperlukan di AWS IoT SiteWise dan AWS IoT Events, Anda dapat mengaktifkan aliran data antara sumber daya untuk mengaktifkan alarm Anda. Di bagian ini, Anda memperbarui definisi alarm dalam model aset untuk menggunakan model alarm yang Anda buat pada langkah sebelumnya.

Untuk mengaktifkan aliran data antara AWS IoT SiteWise dan AWS IoT Events (CLI)

- Atur model alarm sebagai sumber alarm dalam model aset. Lakukan hal-hal berikut:

- a. Jalankan perintah berikut untuk mengambil definisi model aset yang ada. Ganti *asset-model-id* dengan ID model aset.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Operasi mengembalikan respons yang berisi detail model aset.

- b. Buat file bernama `update-asset-model-payload.json` dan salin respons perintah sebelumnya ke dalam file.
- c. Hapus pasangan kunci-nilai berikut dari file: `update-asset-model-payload.json`
- `assetModelId`
 - `assetModelArn`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
- d. Tambahkan properti sumber alarm (`AWS/ALARM_SOURCE`) ke model komposit alarm yang Anda tentukan sebelumnya. Ganti *alarmModelArn* dengan ARN model alarm, yang menetapkan nilai properti sumber alarm.

```
{
  ...
  "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. Jalankan perintah berikut untuk memperbarui model aset dengan definisi yang disimpan dalam `update-asset-model-payload.json` file. Ganti *asset-model-id* dengan ID model aset.

```

aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://update-asset-model-payload.json

```

Model aset Anda sekarang mendefinisikan alarm yang mendeteksi. AWS IoT Events Alarm memantau properti target di semua aset berdasarkan model aset ini. Anda dapat mengonfigurasi alarm pada setiap aset untuk menyesuaikan properti seperti ambang batas atau penerima Pusat Identitas IAM untuk setiap aset. Untuk informasi selengkapnya, lihat [Konfigurasi alarm pada aset di AWS IoT SiteWise](#).

Tentukan alarm eksternal di AWS IoT SiteWise

Alarm eksternal berisi status alarm yang Anda deteksi di luar. AWS IoT SiteWise

Tentukan alarm eksternal (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menentukan alarm eksternal pada model aset yang ada. Untuk menentukan alarm eksternal pada model aset baru, buat model aset, lalu selesaikan langkah-langkah ini. Untuk informasi selengkapnya, lihat [Buat model aset di AWS IoT SiteWise](#).

Untuk menentukan alarm pada model aset

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model.
3. Pilih model aset untuk menentukan alarm.
4. Pilih tab Definisi alarm.
5. Pilih Tambahkan alarm.
6. Di Opsi jenis alarm, pilih Alarm eksternal.
7. Masukkan nama untuk alarm Anda.
8. (Opsional) Masukkan deskripsi untuk alarm Anda.
9. Pilih Tambahkan alarm.

Tentukan alarm eksternal (CLI)

Anda dapat menggunakan AWS CLI untuk menentukan alarm eksternal pada model aset baru atau yang sudah ada.

Untuk menambahkan alarm eksternal ke model aset, Anda menambahkan model komposit alarm ke model aset. Model komposit alarm eksternal menentukan EXTERNAL jenis dan tidak menentukan properti sumber alarm. Contoh berikut alarm komposit mendefinisikan alarm suhu eksternal.

```
{
  ...
  "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": {}
      }
    }
  ]
}
```

Untuk informasi selengkapnya tentang cara menambahkan model komposit ke model aset baru atau yang sudah ada, lihat berikut ini:

- [Buat model aset \(AWS CLI\)](#)
- [Memperbarui model aset atau komponen \(AWS CLI\)](#)

Setelah Anda menentukan alarm eksternal, Anda dapat menelan status alarm ke aset berdasarkan model aset. Untuk informasi selengkapnya, lihat [Menelan status alarm eksternal di AWS IoT SiteWise](#).

Konfigurasi alarm pada aset di AWS IoT SiteWise

Setelah menentukan AWS IoT Events alarm pada model aset, Anda dapat mengonfigurasi alarm pada setiap aset berdasarkan model aset. Anda dapat mengedit nilai ambang batas dan pengaturan notifikasi untuk alarm. Masing-masing nilai ini adalah atribut pada aset, sehingga Anda dapat memperbarui nilai default atribut untuk mengonfigurasi nilai-nilai ini.

 Note

Anda dapat mengonfigurasi nilai-nilai ini untuk AWS IoT Events alarm, tetapi tidak pada alarm eksternal.

Topik

- [Konfigurasi nilai ambang batas \(konsol\)](#)
- [Konfigurasi nilai ambang \(AWS CLI\)](#)
- [Konfigurasi pengaturan notifikasi di AWS IoT SiteWise](#)

Konfigurasi nilai ambang batas (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk memperbarui nilai atribut yang menentukan nilai ambang alarm.

Untuk memperbarui nilai ambang alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda perbarui nilai ambang alarm.

 Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih Edit.
5. Temukan atribut yang digunakan alarm untuk nilai ambangnya, lalu masukkan nilai barunya.
6. Pilih Simpan.

Konfigurasi nilai ambang (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk memperbarui nilai atribut yang menentukan nilai ambang alarm.

Anda harus mengetahui aset `assetId` dan properti Anda `propertyId` untuk menyelesaikan prosedur ini. Anda juga dapat menggunakan ID eksternal. Jika Anda membuat aset dan tidak mengetahuinya `assetId`, gunakan [ListAssets](#) API untuk mencantumkan semua aset untuk model tertentu. Gunakan [DescribeAsset](#) operasi untuk melihat properti aset Anda termasuk properti IDs.

Gunakan [BatchPutAssetPropertyValue](#) operasi untuk menetapkan nilai atribut ke aset Anda. Anda dapat menggunakan operasi ini untuk mengatur beberapa atribut sekaligus. Payload operasi ini berisi daftar entri, dan setiap entri berisi ID aset, ID properti, dan nilai atribut.

Untuk memperbarui nilai atribut (AWS CLI)

1. Buat file bernama `batch-put-payload.json` dan salin objek JSON berikut ke dalam file. Contoh payload ini menunjukkan cara mengatur lintang dan bujur turbin angin. Perbarui IDs, nilai, dan stempel waktu untuk memodifikasi payload untuk kasus penggunaan Anda.

```
{
  "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
      }
    }
  ]
}

```

- Setiap entri dalam payload berisi `entryId` yang dapat Anda definisikan sebagai string unik apa pun. Jika ada entri permintaan yang gagal, setiap kesalahan akan berisi permintaan `entryId` yang sesuai sehingga Anda tahu permintaan mana yang harus dicoba lagi.
- Untuk menetapkan nilai atribut, Anda dapat menyertakan satu timestamp-quality-value (TQV) struktur dalam daftar `propertyValues` untuk setiap properti atribut. Struktur ini harus berisi yang baru `value` dan saat `timestamp`.
 - `value`— Struktur yang berisi salah satu bidang berikut, tergantung pada jenis properti yang ditetapkan:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
 - `timestamp`— Struktur yang berisi waktu epoch Unix saat ini dalam hitungan detik,. `timeInSeconds` AWS IoT SiteWise menolak setiap titik data dengan stempel waktu yang ada lebih dari 7 hari di masa lalu atau lebih baru dari 5 menit di masa depan.

Untuk informasi lebih lanjut tentang cara menyiapkan muatan [BatchPutAssetPropertyValue](#), lihat [Menelan data dengan AWS IoT SiteWise APIs](#).

2. Jalankan perintah berikut untuk mengirim nilai atribut ke AWS IoT SiteWise:

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

Konfigurasi pengaturan notifikasi di AWS IoT SiteWise

Anda dapat mengonfigurasi pengaturan notifikasi alarm menggunakan AWS IoT SiteWise konsol atau AWS Command Line Interface (AWS CLI).

Konfigurasi pengaturan notifikasi (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk memperbarui nilai atribut yang menentukan pengaturan notifikasi untuk alarm.

Untuk memperbarui pengaturan notifikasi alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda perbarui pengaturan alarmnya.
4. Pilih Edit.
5. Temukan atribut yang digunakan alarm untuk pengaturan notifikasi yang ingin Anda ubah, lalu masukkan nilai barunya.
6. Pilih Simpan.

Konfigurasi pengaturan notifikasi (CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk memperbarui nilai atribut yang menentukan pengaturan notifikasi untuk alarm.

Anda harus mengetahui aset `assetId` dan properti Anda `propertyId` untuk menyelesaikan prosedur ini. Anda juga dapat menggunakan ID eksternal. Jika Anda membuat aset dan tidak mengetahuinya `assetId`, gunakan [ListAssets](#) API untuk mencantumkan semua aset untuk model tertentu. Gunakan [DescribeAsset](#) operasi untuk melihat properti aset Anda termasuk properti IDs.

Gunakan [BatchPutAssetPropertyValue](#) operasi untuk menetapkan nilai atribut ke aset Anda. Anda dapat menggunakan operasi ini untuk mengatur beberapa atribut sekaligus. Payload operasi ini berisi daftar entri, dan setiap entri berisi ID aset, ID properti, dan nilai atribut.

Untuk memperbarui nilai atribut (AWS CLI)

1. Buat file bernama `batch-put-payload.json` dan salin objek JSON berikut ke dalam file. Contoh payload ini menunjukkan cara mengatur lintang dan bujur turbin angin. Perbarui IDs, nilai, dan stempel waktu untuk memodifikasi payload untuk kasus penggunaan Anda.

```
{
  "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
          }
        }
      ]
    }
  ]
}
```

- Setiap entri dalam payload berisi `entryId` yang dapat Anda definisikan sebagai string unik apa pun. Jika ada entri permintaan yang gagal, setiap kesalahan akan berisi permintaan `entryId` yang sesuai sehingga Anda tahu permintaan mana yang harus dicoba lagi.
- Untuk menetapkan nilai atribut, Anda dapat menyertakan satu `timestamp-quality-value (TQV)` struktur dalam daftar `propertyValues` untuk setiap properti atribut. Struktur ini harus berisi yang baru `value` dan saat `initimestamp`.

- **value**— Struktur yang berisi salah satu bidang berikut, tergantung pada jenis properti yang ditetapkan:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- **timestamp**— Struktur yang berisi waktu epoch Unix saat ini dalam hitungan detik,. `timeInSeconds` AWS IoT SiteWise menolak setiap titik data dengan stempel waktu yang ada lebih dari 7 hari di masa lalu atau lebih baru dari 5 menit di masa depan.

Untuk informasi lebih lanjut tentang cara menyiapkan muatan [BatchPutAssetPropertyValue](#), lihat [Menelan data dengan AWS IoT SiteWise APIs](#).

2. Jalankan perintah berikut untuk mengirim nilai atribut ke AWS IoT SiteWise:

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

Menanggapi alarm di AWS IoT SiteWise

Ketika AWS IoT Events alarm berubah status, Anda dapat melakukan hal berikut untuk menanggapi alarm:

- Akui alarm untuk menunjukkan bahwa Anda menangani masalah tersebut.
- Tunda alarm untuk menonaktifkannya sementara.
- Nonaktifkan alarm untuk menonaktifkannya secara permanen sampai Anda mengaktifkannya lagi.
- Aktifkan alarm yang dinonaktifkan untuk mendeteksi status alarm.
- Setel ulang alarm untuk menghapus status dan nilai terbarunya.

Anda dapat menggunakan AWS IoT SiteWise konsol atau AWS IoT Events API untuk merespons alarm.

 Note

Anda dapat merespons AWS IoT Events alarm, tetapi bukan alarm eksternal.

Topik

- [Menanggapi alarm \(konsol\)](#)
- [Menanggapi alarm \(API\)](#)

Menanggapi alarm (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk mengakui, menunda, menonaktifkan, atau mengaktifkan alarm.

Topik

- [Akui alarm \(konsol\)](#)
- [Tunda alarm \(konsol\)](#)
- [Nonaktifkan alarm \(konsol\)](#)
- [Aktifkan alarm \(konsol\)](#)
- [Setel ulang alarm \(konsol\)](#)

Akui alarm (konsol)

Anda dapat menerima alarm untuk menunjukkan bahwa Anda menangani masalah tersebut.

 Note

Anda harus mengaktifkan aliran pengakuan pada alarm sehingga Anda dapat mengenali alarm. Opsi ini diaktifkan secara default jika Anda menentukan alarm dari AWS IoT SiteWise konsol.

Untuk mengakui alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda akui alarmnya.

 Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih tab Alarm.
5. Pilih alarm untuk mengakui, lalu pilih Tindakan untuk membuka menu tindakan respons.
6. Pilih Akui. Status alarm berubah menjadi Diakui.

Tunda alarm (konsol)

Anda dapat menunda alarm untuk menonaktifkannya sementara. Tentukan durasi untuk menunda alarm.

Untuk menunda alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda tunda alarm.

 Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih tab Alarm.
5. Pilih alarm untuk ditunda, lalu pilih Tindakan untuk membuka menu tindakan respons.
6. Pilih Tunda. Model terbuka di mana Anda menentukan durasi untuk menunda.
7. Pilih panjang Tunda atau masukkan panjang Tunda khusus.
8. Pilih Simpan. Status alarm berubah menjadi Snoozed.

Nonaktifkan alarm (konsol)

Anda dapat menonaktifkan alarm sehingga tidak mendeteksi lagi. Setelah Anda menonaktifkan alarm, Anda harus mengaktifkannya lagi jika Anda ingin alarm mendeteksi.

Untuk menonaktifkan alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda nonaktifkan alarmnya.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih tab Alarm.
5. Pilih alarm yang akan dinonaktifkan, lalu pilih Tindakan untuk membuka menu tindakan respons.
6. Pilih Disable (Nonaktifkan). Status alarm berubah menjadi Dinonaktifkan.

Aktifkan alarm (konsol)

Anda dapat mengaktifkan alarm untuk mendeteksi lagi setelah Anda menonaktifkan atau menundanya.

Untuk mengaktifkan alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda aktifkan alarmnya.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih tab Alarm.

5. Pilih alarm yang akan diaktifkan, lalu pilih Tindakan untuk membuka menu tindakan respons.
6. Pilih Aktifkan. Status alarm berubah menjadi Normal.

Setel ulang alarm (konsol)

Anda dapat mengatur ulang alarm untuk menghapus status dan nilai terbarunya.

Untuk mengatur ulang alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda atur ulang alarm.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih tab Alarm.
5. Pilih alarm yang akan diaktifkan, lalu pilih Tindakan untuk membuka menu tindakan respons.
6. Pilih Reset. Status alarm berubah menjadi Normal.

Menanggapi alarm (API)

Anda dapat menggunakan AWS IoT Events API untuk mengakui, menunda, menonaktifkan, mengaktifkan, atau mengatur ulang alarm. Untuk informasi selengkapnya, lihat operasi berikut di Referensi AWS IoT Events API:

- [BatchAcknowledgeAlarm](#)
- [BatchSnoozeAlarm](#)
- [BatchDisableAlarm](#)
- [BatchEnableAlarm](#)
- [BatchResetAlarm](#)

Untuk informasi selengkapnya, lihat [Menanggapi alarm](#) di Panduan AWS IoT Events Pengembang.

Menelan status alarm eksternal di AWS IoT SiteWise

Alarm eksternal adalah alarm yang Anda evaluasi di luar. AWS IoT SiteWise Anda dapat menggunakan alarm eksternal ketika Anda memiliki sumber data yang melaporkan status alarm yang ingin Anda konsumsi. AWS IoT SiteWise

Properti status alarm memerlukan format khusus untuk nilai data status alarm. Setiap nilai data harus berupa objek JSON yang diserialisasikan ke string. Kemudian, Anda menelan string serial sebagai nilai string. Untuk informasi selengkapnya, lihat [Properti status alarm](#).

Example Contoh nilai data status alarm (tidak diserialisasi)

```
{
  "stateName": "Active"
}
```

Example Contoh nilai data status alarm (serial)

```
{"stateName\\":\\"Active\\"}
```

Note

Jika sumber data tidak dapat melaporkan data dalam format ini, atau Anda tidak dapat mengonversi data ke format ini sebelum menelannya, Anda dapat memilih untuk tidak menggunakan properti alarm. Sebagai gantinya, Anda dapat menelan data sebagai properti pengukuran dengan tipe data string, misalnya. Untuk informasi selengkapnya, lihat [Tentukan aliran data dari peralatan \(pengukuran\)](#) dan [Menelan data ke AWS IoT SiteWise](#).

Petakan aliran status alarm eksternal di AWS IoT SiteWise

Anda dapat menentukan alias properti untuk memetakan aliran data Anda ke properti status alarm Anda. Ini membantu Anda dengan mudah mengidentifikasi properti status alarm saat Anda menelan atau mengambil data. Untuk informasi selengkapnya tentang alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Topik

- [Memetakan aliran status alarm eksternal \(konsol\)](#)

- [Peta aliran status alarm eksternal \(\)AWS CLI](#)

Memetakan aliran status alarm eksternal (konsol)

Anda dapat menentukan alias properti untuk memetakan aliran data Anda ke properti status alarm Anda. Ini membantu Anda dengan mudah mengidentifikasi properti status alarm saat Anda menelan atau mengambil data. Untuk informasi selengkapnya tentang alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Anda dapat menggunakan AWS IoT SiteWise konsol untuk menyetel alias untuk properti status alarm.

Untuk menyetel alias properti untuk properti status alarm (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset yang ingin Anda tetapkan alias properti.

 Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih Edit.
5. Gulir ke Alarm dan perluas bagian.
6. Di bawah Alarm Eksternal, masukkan alias di alias Properti - opsional.
7. Pilih Simpan.

Peta aliran status alarm eksternal ()AWS CLI

Anda dapat menentukan alias properti untuk memetakan aliran data Anda ke properti status alarm Anda. Ini membantu Anda dengan mudah mengidentifikasi properti status alarm saat Anda menelan atau mengambil data. Untuk informasi selengkapnya tentang alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menyetel alias untuk properti status alarm.

Anda harus mengetahui aset `assetId` dan properti Anda `propertyId` untuk menyelesaikan prosedur ini. Anda juga dapat menggunakan ID eksternal. Jika Anda membuat aset dan tidak mengetahuinya `assetId`, gunakan [ListAssets](#) API untuk mencantumkan semua aset untuk model tertentu. Gunakan [DescribeAsset](#) operasi untuk melihat properti aset Anda termasuk properti IDs.

 Note

[DescribeAsset](#) Tanggapan tersebut mencakup daftar model aset komposit untuk aset tersebut. Setiap alarm adalah model komposit. Untuk menemukan `propertyId`, temukan model komposit untuk alarm, dan kemudian temukan `AWS/ALARM_STATE` properti dalam model komposit itu.

Untuk informasi selengkapnya tentang cara menyetel alias properti, lihat [Memperbarui alias properti aset](#).

Menelan data status alarm di AWS IoT SiteWise

Properti status alarm mengharapkan status alarm sebagai string JSON serial. Untuk menyerap status alarm ke alarm eksternal AWS IoT SiteWise, Anda menelan string serial ini sebagai nilai string stempel waktu. Contoh berikut menunjukkan nilai data status untuk alarm aktif.

```
{\"stateName\":\"Active\"}
```

Untuk mengidentifikasi properti status alarm, Anda dapat menentukan salah satu dari berikut ini:

- Alarm `assetId` dan `propertyId` properti alarm yang Anda kirim datanya.
- The `propertyAlias`, yang merupakan alias aliran data (misalnya, `/company/windfarm/3/turbine/7/temperature/high`). Untuk menggunakan opsi ini, Anda harus terlebih dahulu mengatur alias properti alarm Anda. Untuk mempelajari cara menyetel alias properti untuk properti status alarm, lihat [Petakan aliran status alarm eksternal di AWS IoT SiteWise](#).

Contoh payload [BatchPutAssetPropertyValue](#) API berikut menunjukkan cara memformat status alarm eksternal. Alarm eksternal ini melaporkan ketika pembacaan rotasi turbin angin per menit (RPM) terlalu tinggi.

Example Contoh BatchPutAssetPropertyValue payload untuk data status alarm

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\":\"Active\"}"
          },
          "timestamp": {
            "timeInSeconds": 1607550262
          }
        }
      ]
    }
  ]
}
```

Untuk informasi selengkapnya tentang cara menggunakan BatchPutAssetPropertyValue API untuk menyerap data, lihat [Menelan data dengan AWS IoT SiteWise APIs](#).

Untuk informasi selengkapnya tentang cara lain untuk menelan data, lihat [Menelan data ke AWS IoT SiteWise](#).

AWS IoT SiteWise Asisten

AWS IoT SiteWise Asisten adalah asisten bertenaga AI generatif. Ini memungkinkan pengguna seperti manajer pabrik, insinyur berkualitas, dan teknisi pemeliharaan untuk mendapatkan wawasan, memecahkan masalah, dan mengambil tindakan langsung dari data operasional dan perusahaan mereka.

AWS IoT SiteWise Asisten mengkonsolidasikan informasi dari AWS IoT data, model aset, manual, dan dokumentasi ke dalam ringkasan peristiwa kritis yang dapat dimengerti. Ini juga memungkinkan sesi tanya jawab mendalam interaktif untuk diagnosis yang mudah, eksplorasi akar penyebab dan rekomendasi terpandu.

Topik

- [Konfigurasi AWS IoT SiteWise Asisten](#)
- [Buat kumpulan data](#)
- [Mengedit kumpulan data](#)
- [Menghapus kumpulan data](#)
- [AWS IoT SiteWise Pertanyaan asisten](#)

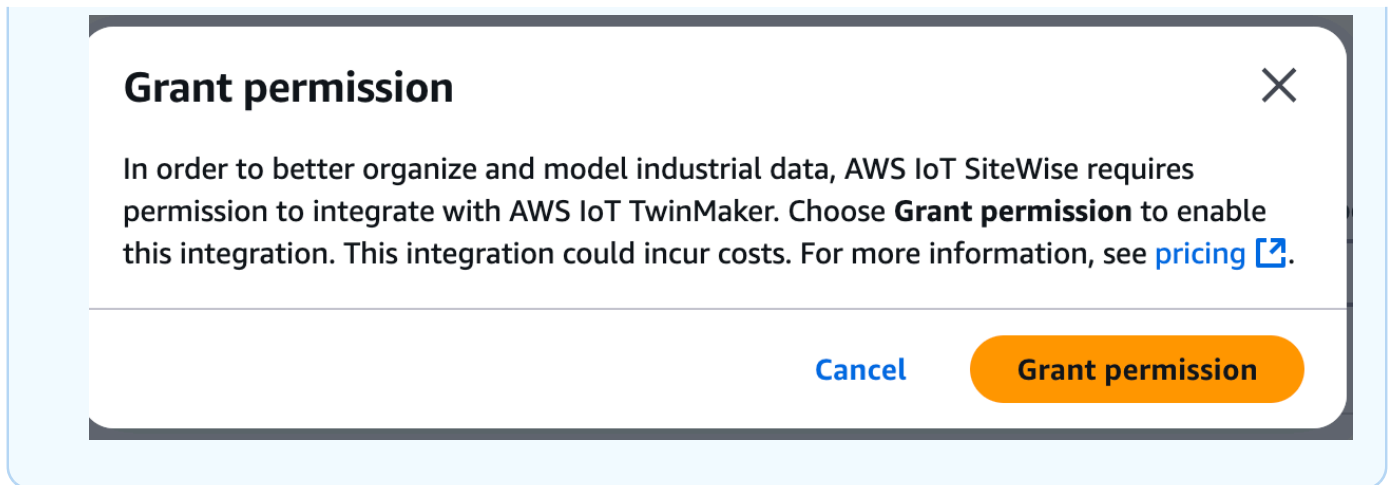
Konfigurasi AWS IoT SiteWise Asisten

AWS IoT SiteWise Konfigurasi asisten

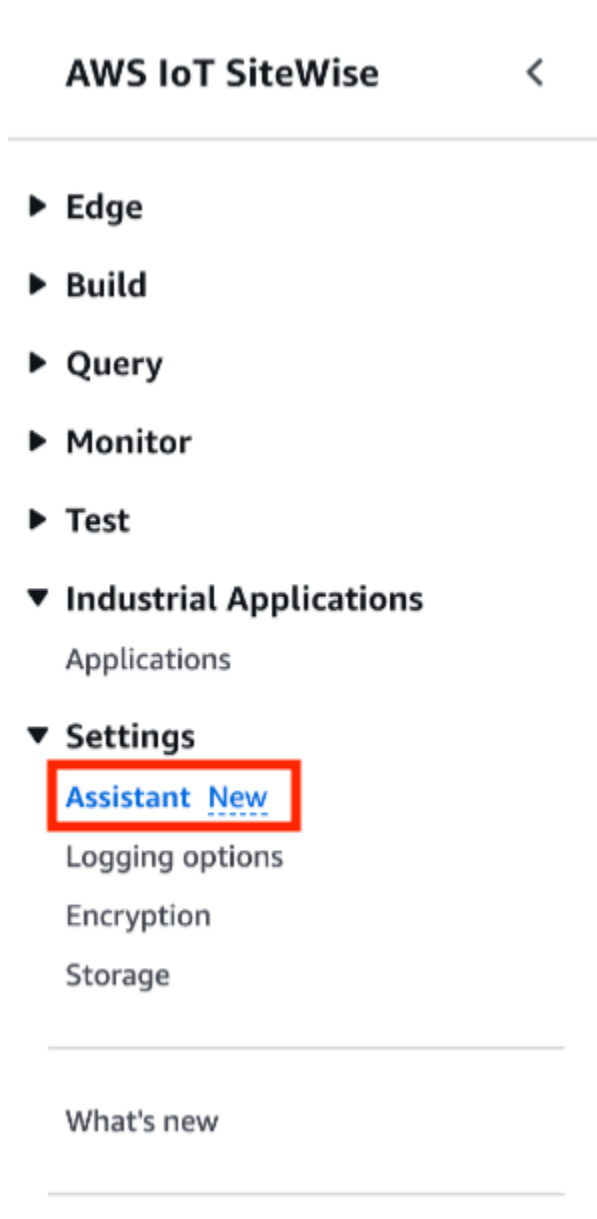
1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.

Note

Berikan izin untuk mengaktifkan integrasi dengan AWS IoT TwinMaker layanan. Ini diperlukan untuk AWS IoT SiteWise Asisten, dan dasbor untuk mengeksekusi kueri SQL dalam AWS IoT SiteWise sumber daya. Lihat [Integrasi AWS IoT SiteWise dan AWS IoT TwinMaker](#).



2. Pilih Asisten dari panel navigasi kiri.



Buat kumpulan data

Note

AWS IoT SiteWise Asisten harus menggunakan kumpulan data dengan indeks [Amazon Kendra](#) untuk pengetahuan dan panduan tingkat perusahaan. Jika Anda tidak memiliki indeks Amazon Kendra, lihat [Membuat indeks](#) untuk membuatnya. Menambahkan [kumpulan data](#) meningkatkan kualitas respons Asisten, dan meminimalkan halusinasi.

Console

Buat kumpulan data di konsol AWS IoT SiteWise

1. Dataset ditampilkan di bagian Datasets dari halaman Asisten.AWS IoT SiteWise
2. Jika tidak ada kumpulan data, pilih Buat kumpulan data.
3. Di halaman detail Dataset, pilih indeks Kendra dari menu tarik-turun untuk dikaitkan dengan kumpulan data.
4. Nama dataset diisi oleh indeks Kendra yang dipilih pada Langkah 3. Edit nama jika diperlukan.
5. (Opsional) Deskripsi kumpulan data diisi oleh indeks Kendra yang dipilih pada Langkah 3. Edit deskripsi jika diperlukan.
6. Di bagian Izin, pilih dari bawah ini:
 - a. Pilih Buat dan gunakan peran layanan baru. Secara default, AWS IoT SiteWise secara otomatis membuat peran layanan. Peran ini memungkinkan AWS IoT SiteWise Asisten untuk mengakses indeks Kendra Anda.
 - b. Pilih Gunakan peran layanan yang ada, lalu pilih peran target.
7. Pilih Buat.

[AWS IoT SiteWise](#) > [Assistant](#) > Create dataset

Create dataset

Create a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index

[Amazon Kendra](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#).

Choose a method to authorize Assistant

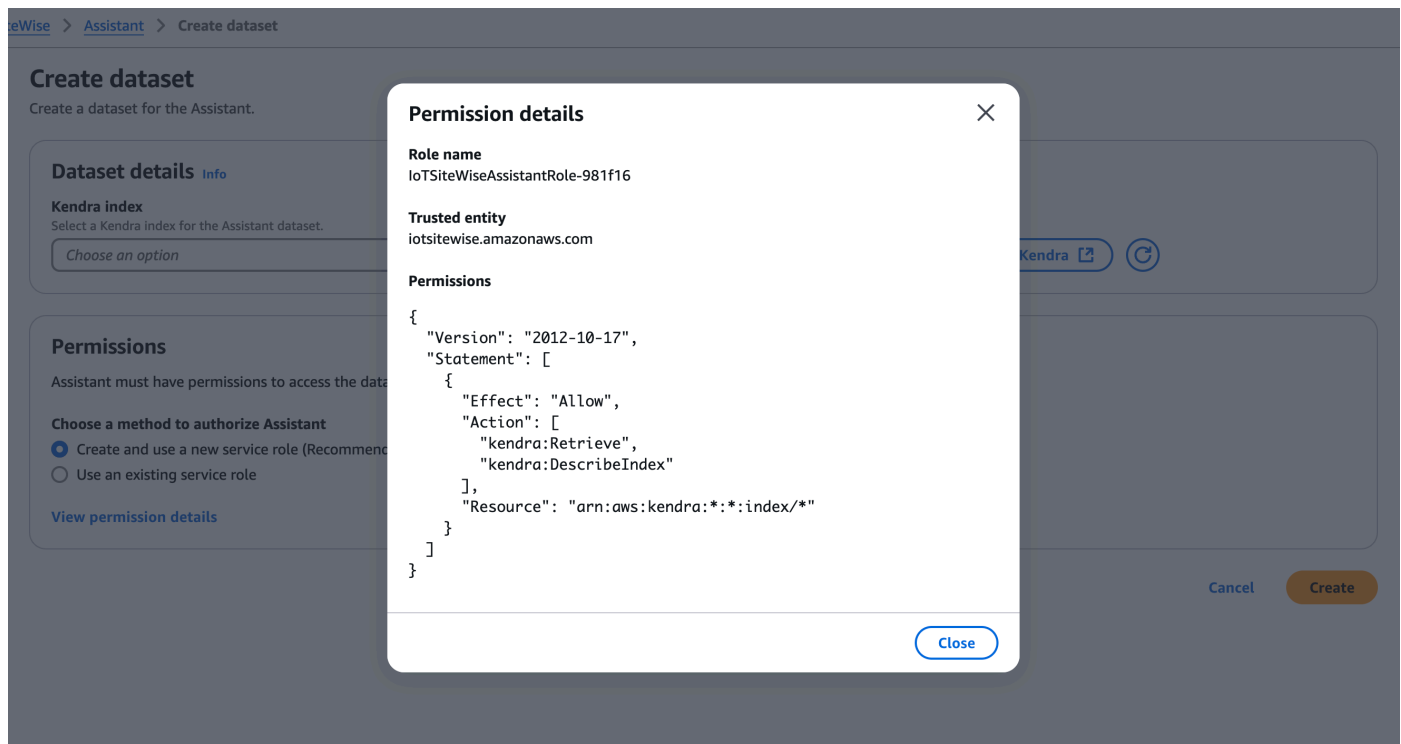
- ☒ Create and use a new service role - *recommended*
- ☐ Use an existing service role

[View permission details](#)

[Cancel](#)

Create

Peran layanan yang dibuat oleh AWS IoT SiteWise untuk pengguna, jika pengguna memilih untuk Membuat dan menggunakan peran layanan baru.



AWS CLI

Buat kumpulan data di AWS CLI

1. Buat peran IAM yang digunakan untuk membuat kumpulan data. Gunakan kebijakan izin berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kendra:Retrieve"
      ],
      "Resource": "arn:aws:kendra:*:*:index/*"
    }
  ]
}
```

Gunakan hubungan kepercayaan berikut:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Buat file `create-dataset.json` dengan template yang disediakan dalam contoh. Isi `datasetId`, `kendra knowledgeBaseArn` dan `roleArn` untuk terhubung dengan dataset ini.

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

3. Buat dataset dengan perintah berikut:

```
aws iotsitewise create-dataset --cli-input-json file://create-dataset.json --
region us-east-1
```

Mengedit kumpulan data

Console

Mengedit kumpulan data

1. Dataset ditampilkan di bagian Datasets dari halaman Asisten. Pilih kumpulan data yang akan diedit. Pilih Edit untuk mulai mengedit.
2. Di halaman detail Dataset, pilih indeks Kendra dari menu tarik-turun untuk dikaitkan dengan kumpulan data.
3. Nama dataset diisi oleh indeks Kendra yang dipilih pada Langkah 2. Edit nama jika diperlukan.
4. (Opsional) Deskripsi kumpulan data diisi oleh indeks Kendra yang dipilih pada Langkah 2. Edit deskripsi jika diperlukan.
5. Di bagian Izin, pilih dari bawah ini:
 - a. Pilih Buat dan gunakan peran layanan baru. Secara default, AWS IoT SiteWise secara otomatis membuat peran layanan. Peran ini memungkinkan AWS IoT SiteWise Asisten untuk mengakses indeks Kendra Anda.
 - b. Pilih Gunakan peran layanan yang ada, lalu pilih peran target.
6. Pilih Simpan perubahan untuk menyimpan pilihan Anda.

[AWS IoT SiteWise](#) > [Assistant](#) > Edit dataset

Edit dataset

Edit a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index

[Amazon Kendra](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#).

Choose a method to authorize Assistant

- ☐ Create and use a new service role - *recommended*
- ☒ Use an existing service role

Existing role

IoTSiteWiseAssistantRole-40f994

[View the IoTSiteWiseAssistantRole-40f994 role in the IAM console](#)

[Cancel](#)

[Save changes](#)

AWS CLI

Mengedit kumpulan data di AWS CLI

1. Buat file update-dataset.json dengan template yang disediakan dalam contoh. IsidatasetId, kendra knowledgeBaseArn dan roleArn untuk terhubung dengan dataset ini.

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```



```
}  
  }  
} }
```

2. Perbarui dataset dengan perintah berikut:

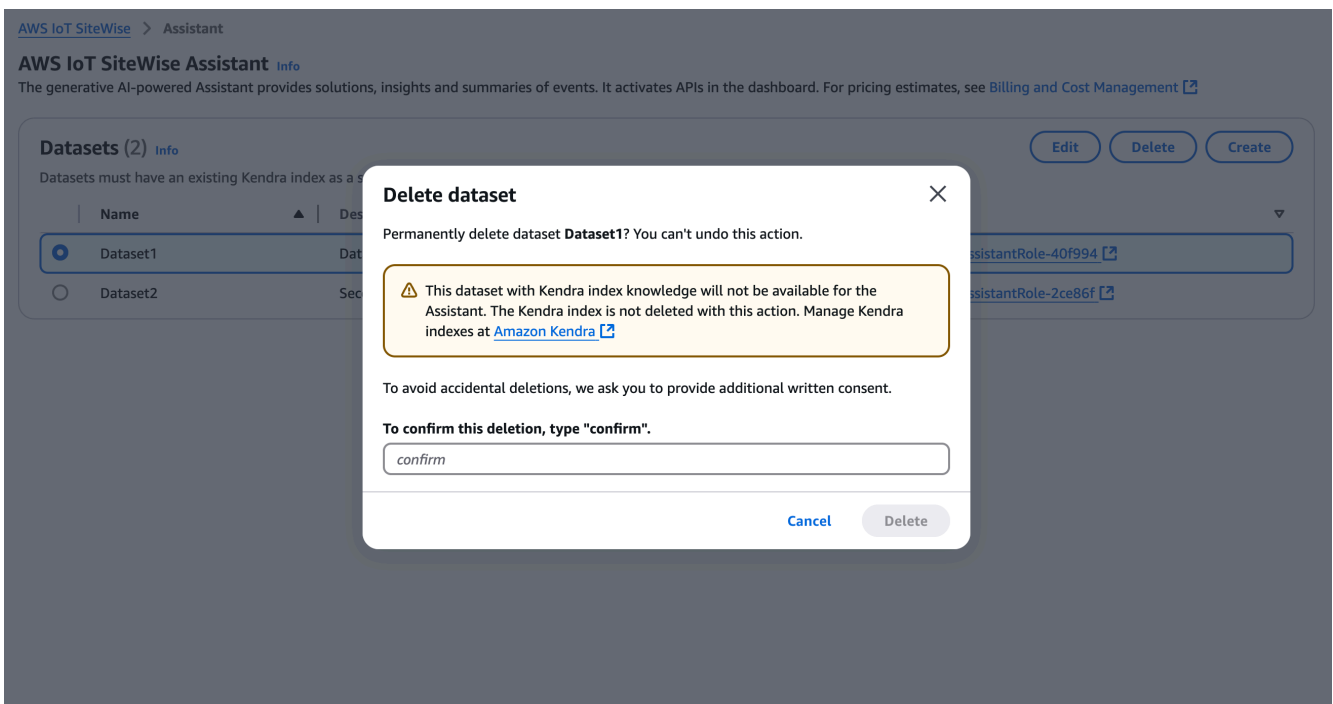
```
aws iotsitewise update-dataset --cli-input-json file://update-dataset.json --  
region us-east-1
```

Menghapus kumpulan data

Console

Menghapus kumpulan data

1. Dataset ditampilkan di bagian Datasets dari halaman Asisten. Pilih kumpulan data. Pilih Hapus.
2. Ketik konfirmasi di popup untuk mengonfirmasi penghapusan.



3. Pilih Hapus.

AWS CLI

Menghapus kumpulan data

- Hapus kumpulan data dengandatasetId.

```
aws iotsitewise delete-dataset --region us-east-1 --dataset-id <UUID>
```

AWS IoT SiteWise Pertanyaan asisten

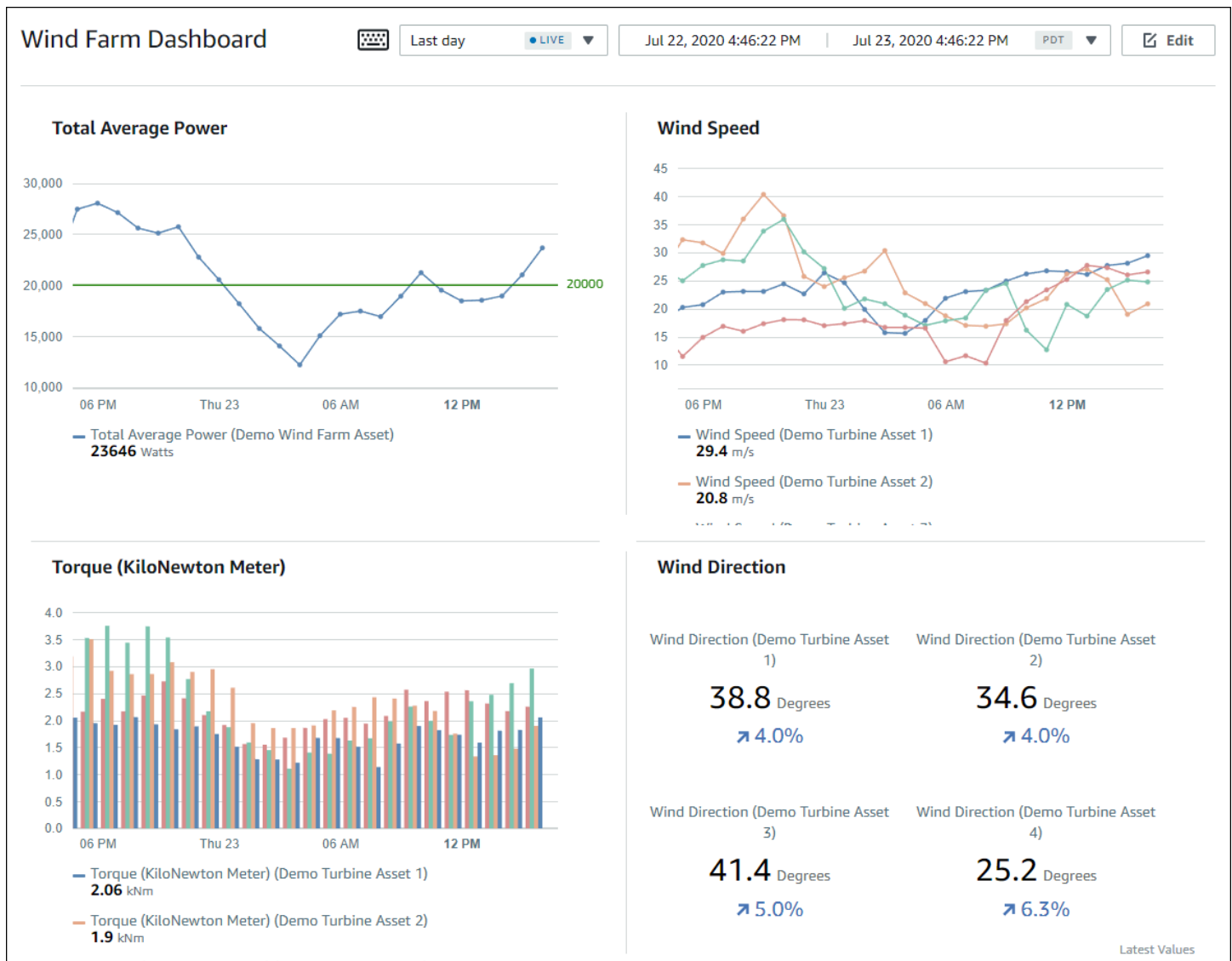
Lihat [Contoh pertanyaan untuk ditanyakan kepada AWS IoT SiteWise Asisten](#) untuk mempelajari lebih lanjut tentang menanyakan AWS IoT SiteWise Asisten.

Memantau data dengan AWS IoT SiteWise Monitor

Anda dapat menggunakan AWS IoT SiteWise untuk memantau data dari proses, perangkat, dan peralatan Anda dengan membuat portal web SiteWise Monitor. SiteWise Monitor adalah fitur AWS IoT SiteWise yang dapat Anda gunakan untuk membuat portal dalam bentuk aplikasi web yang dikelola. Anda kemudian dapat menggunakan portal ini untuk melihat dan membagikan data operasional Anda. Anda dapat membuat proyek dengan dasbor untuk memvisualisasikan data dari proses, perangkat, dan peralatan yang terhubung. AWS IoT

Pakar domain, seperti insinyur proses, dapat menggunakan portal ini untuk dengan cepat mendapatkan wawasan tentang data operasional mereka untuk memahami perilaku perangkat dan peralatan.

Berikut ini adalah contoh dasbor yang menampilkan data untuk ladang angin.



Karena AWS IoT SiteWise menangkap data dari waktu ke waktu, Anda dapat menggunakan SiteWise Monitor untuk melihat data operasional dari waktu ke waktu, atau nilai terakhir yang dilaporkan pada titik waktu tertentu. Ini memungkinkan Anda mengungkap wawasan yang mungkin sulit ditemukan.

SiteWise Memantau peran

Empat peran berinteraksi dengan SiteWise Monitor:

AWS administrator

AWS Administrator menggunakan AWS IoT SiteWise konsol untuk membuat portal. AWS Administrator juga dapat menetapkan administrator portal dan menambahkan pengguna portal.

Administrator portal kemudian menetapkan pengguna portal ke proyek sebagai pemilik atau pemirsa. AWS Administrator bekerja secara eksklusif di AWS konsol.

Administrator portal

Setiap portal SiteWise Monitor memiliki satu atau lebih administrator portal. Administrator portal menggunakan portal untuk membuat proyek yang berisi koleksi aset dan dasbor. Administrator portal kemudian menetapkan aset dan pemilik untuk setiap proyek. Dengan mengontrol akses ke proyek, administrator portal menentukan aset mana yang dapat dilihat oleh pemilik proyek dan pemirsa.

Pemilik proyek

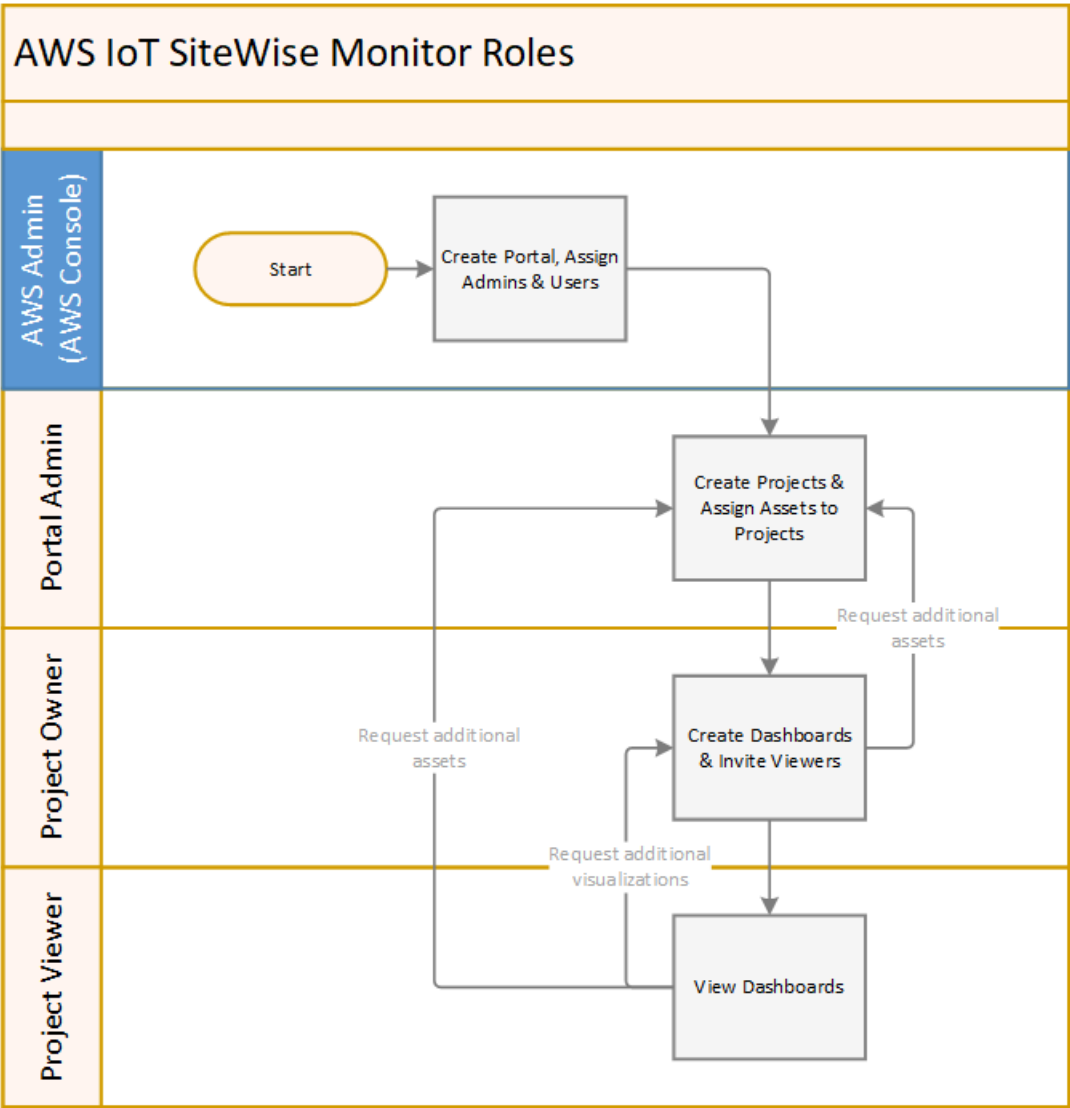
Setiap proyek SiteWise Monitor memiliki pemilik. Pemilik proyek membuat visualisasi dalam bentuk dasbor untuk mewakili data operasional secara konsisten. Ketika dasbor siap untuk dibagikan, pemilik proyek dapat mengundang pemirsa ke proyek. Pemilik proyek juga dapat menugaskan pemilik lain ke proyek. Pemilik proyek dapat mengonfigurasi ambang batas dan pengaturan notifikasi untuk alarm.

Penampil proyek

Setiap proyek SiteWise Monitor memiliki pemirsa. Pemirsa proyek dapat terhubung ke portal untuk melihat dasbor yang dibuat oleh pemilik proyek. Di setiap dasbor, pemirsa proyek dapat menyesuaikan rentang waktu untuk lebih memahami data operasional. Pemirsa proyek hanya dapat melihat dasbor di proyek yang dapat mereka akses. Pemirsa proyek dapat mengenali dan menunda alarm.

Bergantung pada organisasi Anda, orang yang sama mungkin melakukan banyak peran.

Gambar berikut menggambarkan bagaimana keempat peran ini berinteraksi di portal SiteWise Monitor.



Anda dapat mengelola siapa yang memiliki akses ke data Anda dengan menggunakan AWS IAM Identity Center atau IAM. Pengguna data Anda dapat masuk ke SiteWise Monitor dari desktop atau browser seluler menggunakan IAM Identity Center atau IAM mereka.

Federasi SAFL

IAM Identity Center dan IAM mendukung federasi identitas dengan [SAFL \(Security Assertion Markup Language\) 2.0](#). SAFL 2.0 adalah standar terbuka yang digunakan oleh banyak penyedia identitas eksternal (IdPs) untuk mengautentikasi pengguna dan meneruskan identitas dan informasi keamanan mereka ke penyedia layanan (SPs). SPs Biasanya aplikasi atau layanan. Federasi SAFL memungkinkan administrator dan pengguna portal SiteWise Monitor Anda untuk masuk ke portal yang ditetapkan dengan kredensi eksternal, seperti nama pengguna dan kata sandi perusahaan mereka.

Anda dapat mengonfigurasi IAM Identity Center dan IAM untuk menggunakan federasi berbasis SAML untuk akses ke portal Monitor Anda. SiteWise

Pusat Identitas IAM

Administrator dan pengguna portal Anda dapat masuk ke portal AWS akses dengan nama pengguna dan kata sandi perusahaan mereka. Mereka kemudian dapat menavigasi ke portal SiteWise Monitor yang ditugaskan. IAM Identity Center menggunakan sertifikat untuk mengatur hubungan kepercayaan SAFL antara penyedia identitas Anda dan AWS. Untuk informasi lebih lanjut, [profil SCIM dan implementasi SAFL 2.0](#) di AWS IAM Identity Center Panduan Pengguna.

IAM

Administrator dan pengguna portal Anda dapat meminta kredensial keamanan sementara untuk mengakses portal Monitor yang ditetapkan SiteWise. Anda membuat identitas penyedia identitas SALL di IAM untuk mengatur hubungan kepercayaan antara penyedia identitas Anda dan AWS. Untuk informasi selengkapnya, lihat [Menggunakan federasi berbasis SAML untuk akses API AWS](#), di Panduan Pengguna IAM.

Administrator dan pengguna portal Anda dapat masuk ke portal perusahaan Anda dan memilih opsi untuk pergi ke konsol AWS Manajemen. Mereka kemudian dapat menavigasi ke portal SiteWise Monitor yang ditugaskan. Portal perusahaan Anda menangani pertukaran kepercayaan antara penyedia identitas Anda dan AWS. Untuk informasi selengkapnya, lihat [Mengaktifkan pengguna federasi SAFL 2.0 untuk mengakses Konsol AWS Manajemen di Panduan](#) Pengguna IAM.

Note

Saat menambahkan pengguna atau administrator ke portal, hindari membuat kebijakan IAM yang membatasi izin pengguna, seperti IP terbatas. Kebijakan terlampir dengan izin terbatas tidak akan dapat terhubung ke AWS IoT SiteWise portal.

SiteWise Pantau konsep

Untuk menggunakan SiteWise Monitor, Anda harus terbiasa dengan konsep-konsep berikut:

Portal

AWS IoT SiteWise Monitor Portal adalah aplikasi web yang dapat Anda gunakan untuk memvisualisasikan dan berbagi AWS IoT SiteWise data Anda. Portal memiliki satu atau lebih administrator dan berisi nol atau lebih proyek.

Proyek

Setiap portal SiteWise Monitor berisi serangkaian proyek. Setiap proyek memiliki subset AWS IoT SiteWise aset Anda yang terkait dengannya. Pemilik proyek membuat satu atau beberapa dasbor untuk menyediakan cara yang konsisten untuk melihat data yang terkait dengan aset tersebut. Pemilik proyek dapat mengundang pemirsa ke proyek untuk memungkinkan mereka melihat aset dan dasbor dalam proyek. Proyek ini adalah unit dasar berbagi dalam SiteWise Monitor. Pemilik proyek dapat mengundang pengguna yang diberi akses ke portal oleh AWS administrator. Pengguna harus memiliki akses ke portal sebelum proyek di portal itu dapat dibagikan dengan pengguna tersebut.

Aset

Ketika data dicerna AWS IoT SiteWise dari peralatan industri Anda, perangkat, peralatan, dan proses Anda masing-masing direpresentasikan sebagai aset. Setiap aset memiliki properti dan alarm yang terkait dengannya. Administrator portal menetapkan set aset untuk setiap proyek.

Properti

Properti adalah data deret waktu yang terkait dengan aset. Misalnya, peralatan mungkin memiliki nomor seri, lokasi, merek dan model, dan tanggal pemasangan. Mungkin juga memiliki nilai deret waktu untuk ketersediaan, kinerja, kualitas, suhu, tekanan, dan sebagainya.

Alarm

Alarm memantau properti untuk mengidentifikasi kapan peralatan berada di luar jangkauan operasinya. Setiap alarm mendefinisikan ambang batas dan properti untuk dipantau. Ketika properti melebihi ambang batas, alarm menjadi aktif dan menunjukkan bahwa Anda atau seseorang di tim Anda harus mengatasi masalah tersebut. Pemilik proyek dapat menyesuaikan ambang batas dan pengaturan notifikasi untuk alarm. Pemirsa proyek dapat mengenali dan menunda alarm, dan mereka dapat meninggalkan pesan dengan detail tentang alarm atau tindakan yang mereka ambil untuk mengatasinya.

Dasbor

Setiap proyek berisi satu set dasbor. Dasbor menyediakan satu set visualisasi untuk nilai-nilai dari satu set aset. Pemilik proyek membuat dasbor dan visualisasi yang dikandungnya. Ketika

pemilik proyek siap untuk membagikan kumpulan dasbor, pemilik dapat mengundang pemirsa ke proyek, yang memberi mereka akses ke semua dasbor dalam proyek. Jika Anda menginginkan kumpulan pemirsa yang berbeda untuk dasbor yang berbeda, Anda harus membagi dasbor di antara proyek. Saat pemirsa melihat dasbor, mereka dapat menyesuaikan rentang waktu untuk melihat data tertentu.

Visualisasi

Di setiap dasbor, pemilik proyek memutuskan bagaimana menampilkan properti dan alarm aset yang terkait dengan proyek. Ketersediaan dapat direpresentasikan sebagai grafik garis, sementara nilai lainnya mungkin ditampilkan sebagai grafik batang atau indikator kinerja utama (KPIs). Alarm paling baik ditampilkan sebagai kisi-kisi status dan jadwal status. Pemilik proyek menyesuaikan setiap visualisasi untuk memberikan pemahaman terbaik tentang data untuk aset tersebut.

Memulai dengan AWS IoT SiteWise Monitor (Klasik)

Jika Anda AWS administrator untuk organisasi Anda, Anda membuat portal dari AWS IoT SiteWise konsol. Selesaikan langkah-langkah berikut untuk membuat portal sehingga anggota organisasi Anda dapat melihat AWS IoT SiteWise data Anda:

1. Konfigurasi dan buat portal
2. Tambahkan administrator portal dan kirim email undangan
3. Tambahkan pengguna portal

Setelah Anda membuat portal, administrator portal dapat melihat AWS IoT SiteWise aset Anda dan menentukannya ke proyek di portal. Pemilik proyek kemudian dapat membuat dasbor untuk memvisualisasikan properti aset yang membantu pemirsa proyek memahami kinerja perangkat, proses, dan peralatan Anda.

Note

Saat menambahkan pengguna atau administrator ke portal, hindari membuat kebijakan AWS Identity and Access Management (IAM) yang membatasi izin pengguna, seperti IP terbatas. Kebijakan terlampir dengan izin terbatas tidak akan dapat terhubung ke AWS IoT SiteWise portal.

Anda dapat mengikuti tutorial yang berjalan melalui langkah-langkah yang diperlukan untuk menyiapkan portal dengan proyek, dasbor, dan beberapa pengguna untuk skenario tertentu menggunakan data ladang angin. Untuk informasi selengkapnya, lihat [Visualisasikan dan bagikan data ladang angin di Monitor SiteWise](#).

Topik

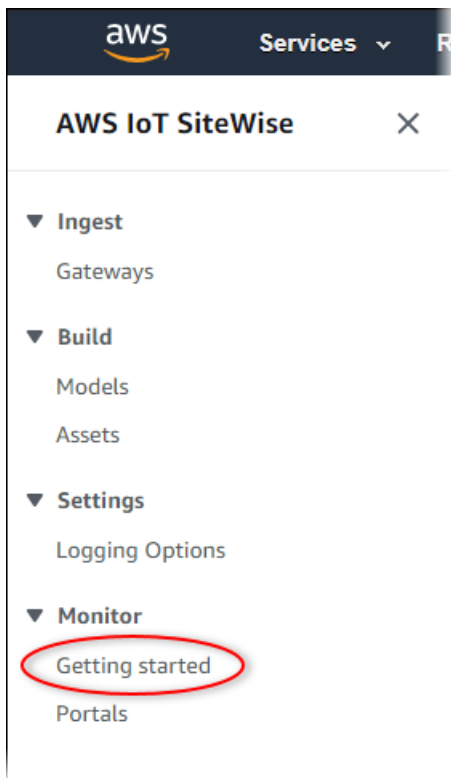
- [Buat portal di SiteWise Monitor](#)
- [Konfigurasi portal Anda di SiteWise Monitor](#)
- [Mengundang administrator di Monitor SiteWise](#)
- [Tambahkan pengguna portal di SiteWise Monitor](#)
- [Buat AWS IoT SiteWise dasbor \(\)AWS CLI](#)
- [Nyalakan alarm untuk portal Anda AWS IoT SiteWise](#)
- [Mengaktifkan AWS IoT SiteWise portal Anda di tepi](#)
- [Mengelola portal SiteWise Monitor Anda](#)

Buat portal di SiteWise Monitor

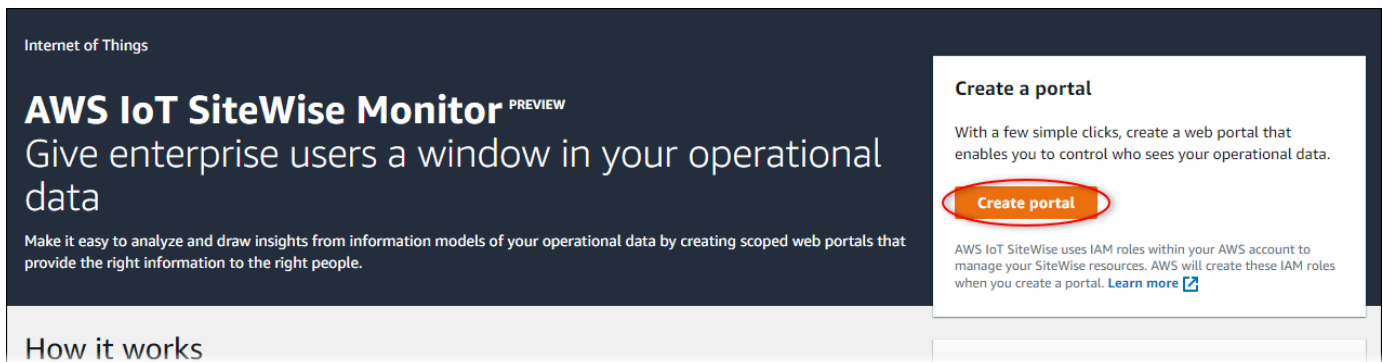
Anda membuat portal SiteWise Monitor di AWS IoT SiteWise konsol.

Untuk membuat portal

1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Monitor, Memulai.



3. Pilih Buat Portal.



Selanjutnya, Anda harus memberikan beberapa informasi dasar untuk mengkonfigurasi portal Anda.

Konfigurasi portal Anda di SiteWise Monitor

Pengguna Anda menggunakan portal untuk melihat data Anda. Anda dapat menyesuaikan nama portal, deskripsi, branding, otentikasi pengguna, email kontak dukungan, dan izin.

AWS IoT SiteWise > Monitor > Portals > 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.

Name should be 1-128 characters and only contain A-Z a-z 0-9 _ and -.

Description - optional

Create a description of your portal

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.

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.

☒ 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

Tags

This resource doesn't have any tags.

You can add up to 50 more tags.

Permissions

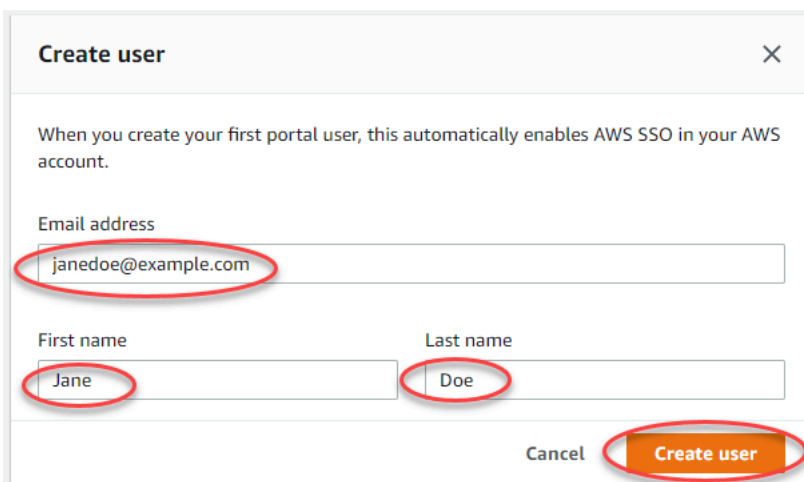
SiteWise Monitor assumes this role to give permissions to your federated users to access AWS IoT SiteWise resources. [Learn](#)

Untuk mengkonfigurasi portal

1. Masukkan nama untuk portal Anda.
2. (Opsional) Masukkan deskripsi untuk portal Anda. Jika Anda memiliki beberapa portal, gunakan deskripsi yang bermakna untuk membantu Anda melacak isi setiap portal.
3. (Opsional) Unggah gambar untuk menampilkan merek Anda di portal. Pilih persegi, gambar PNG. Jika Anda mengunggah gambar non-persegi, portal menskalakan gambar menjadi persegi.
4. Pilih salah satu opsi berikut:
 - Pilih Pusat Identitas IAM jika pengguna portal Anda masuk ke portal ini dengan nama pengguna dan kata sandi perusahaan mereka.

Jika Anda belum mengaktifkan Pusat Identitas IAM di akun Anda, lakukan hal berikut:

- a. Pilih Create user (Buat pengguna).
- b. Pada halaman Buat pengguna, untuk membuat portal pertama, masukkan alamat email pengguna, nama depan, dan nama belakang, lalu pilih Buat pengguna.



Note

- AWS secara otomatis mengaktifkan Pusat Identitas IAM di akun Anda saat Anda membuat pengguna portal pertama.
- Anda dapat mengonfigurasi Pusat Identitas IAM hanya dalam satu Wilayah pada satu waktu. SiteWise Monitor terhubung ke Wilayah yang Anda konfigurasikan untuk Pusat Identitas IAM. Ini berarti Anda menggunakan satu

Wilayah untuk akses Pusat Identitas IAM, tetapi Anda dapat membuat portal di Wilayah mana pun.

- Pilih IAM jika pengguna portal Anda masuk ke portal ini dengan kredensi IAM mereka.

 Important

Pengguna atau peran harus memiliki `iotsitewise:DescribePortal` izin untuk masuk ke portal.

5. Masukkan alamat email yang dapat dihubungi pengguna portal ketika mereka memiliki masalah dengan portal dan perlu bantuan untuk menyelesaikannya.
6. (Opsional) Tambahkan tag untuk portal Anda. Untuk informasi selengkapnya, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).
7. Pilih salah satu opsi berikut:
 - Pilih Buat dan gunakan peran layanan baru. Secara default, SiteWise Monitor secara otomatis membuat peran layanan untuk setiap portal. Peran ini memungkinkan pengguna portal Anda mengakses AWS IoT SiteWise sumber daya Anda. Untuk informasi selengkapnya, lihat [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#).
 - Pilih Gunakan peran layanan yang ada, lalu pilih peran target.
8. Pilih Berikutnya
9. (Opsional) Aktifkan alarm untuk portal Anda. Untuk informasi selengkapnya, lihat [Nyalakan alarm untuk portal Anda AWS IoT SiteWise](#).
10. Pilih Buat. AWS IoT SiteWise akan membuat portal Anda.

 Note

Jika Anda menutup konsol, Anda dapat menyelesaikan proses penyiapan dengan menambahkan administrator dan pengguna. Untuk informasi selengkapnya, lihat [Menambah atau menghapus administrator portal di AWS IoT SiteWise](#). Jika Anda tidak ingin menyimpan portal ini, hapus agar tidak menggunakan sumber daya. Untuk informasi selengkapnya, lihat [Hapus portal di AWS IoT SiteWise](#).

Kolom Status dapat menjadi salah satu nilai berikut.

- **AWS IoT SiteWise CREATING** - sedang memproses permintaan Anda untuk membuat portal. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- **MEMPERBARUI** - AWS IoT SiteWise sedang memproses permintaan Anda untuk memperbarui portal. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- **PENDING** - AWS IoT SiteWise sedang menunggu propagasi catatan DNS selesai. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya. Anda dapat menghapus portal saat statusnya **TERTUNDA**.
- **MENGHAPUS** - AWS IoT SiteWise sedang memproses permintaan Anda untuk menghapus portal. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- **AKTIF** - Ketika portal menjadi aktif, pengguna portal Anda dapat mengaksesnya.
- **GAGAL** - AWS IoT SiteWise tidak dapat memproses permintaan Anda untuk membuat, memperbarui, atau menghapus portal. Jika Anda mengaktifkan AWS IoT SiteWise untuk mengirim log ke Amazon CloudWatch Logs, Anda dapat menggunakan log ini untuk memecahkan masalah. Untuk informasi selengkapnya, lihat [Pemantauan AWS IoT SiteWise dengan CloudWatch Log](#).

Pesan muncul saat portal Anda dibuat.

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-1111EXAMPLE.app.iotsitewise.aws".

Successfully created portal URL at <https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws>

Selanjutnya, Anda harus mengundang satu atau lebih administrator portal ke portal. Sejauh ini, Anda membuat portal tetapi tidak ada yang bisa mengaksesnya.

Mengundang administrator di Monitor SiteWise

Untuk memulai di portal baru Anda, Anda harus menetapkan administrator portal. Administrator portal membuat proyek, memilih pemilik proyek, dan menetapkan aset untuk proyek. Administrator portal dapat melihat semua AWS IoT SiteWise aset Anda.

Berdasarkan layanan otentikasi pengguna, pilih salah satu opsi berikut:

IAM Identity Center

Jika Anda menggunakan SiteWise Monitor untuk pertama kalinya, Anda dapat memilih pengguna yang Anda buat sebelumnya untuk menjadi administrator portal. Jika Anda ingin menambahkan pengguna lain sebagai administrator portal, Anda dapat membuat pengguna Pusat Identitas IAM

dari halaman ini. Atau, Anda dapat menghubungkan penyedia identitas eksternal ke IAM Identity Center. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS IAM Identity Center](#).

Untuk mengundang administrator

1. Pilih kotak centang untuk pengguna yang Anda inginkan sebagai administrator portal Anda. Ini menambahkan pengguna ke daftar administrator Portal.

 Note

Jika Anda menggunakan Pusat Identitas IAM sebagai penyimpanan identitas, dan Anda masuk ke akun AWS Organizations manajemen, Anda dapat memilih Buat pengguna untuk membuat pengguna Pusat Identitas IAM. IAM Identity Center mengirimkan email kepada pengguna baru agar mereka dapat mengatur kata sandi mereka. Anda kemudian dapat menetapkan pengguna ke portal sebagai administrator. Untuk informasi selengkapnya, lihat [Mengelola identitas di Pusat Identitas IAM](#).

2. (Opsional) Pilih Kirim undangan ke pengguna yang dipilih. Klien email Anda terbuka, dan undangan diisi di badan pesan.

Anda dapat menyesuaikan email sebelum mengirimkannya ke administrator portal Anda. Anda juga dapat mengirim email ke administrator portal Anda nanti. Jika Anda mencoba SiteWise Monitor untuk pertama kalinya dan menambahkan Pusat Identitas IAM baru atau pengguna IAM atau peran sebagai administrator portal, Anda tidak perlu mengirim email sendiri.

3. Jika Anda menambahkan pengguna yang tidak Anda inginkan sebagai administrator, kosongkan kotak centang untuk pengguna tersebut.
4. Setelah selesai mengundang administrator portal, pilih Berikutnya.

IAM

Anda dapat memilih pengguna atau peran untuk menjadi administrator portal. Jika ingin menambahkan pengguna atau peran lain sebagai administrator portal, Anda dapat membuat pengguna atau peran di konsol IAM. Untuk informasi selengkapnya, lihat [Membuat pengguna IAM di AWS akun Anda](#) dan [Membuat peran IAM](#) di Panduan Pengguna IAM.

Untuk mengundang administrator

1. Lakukan hal-hal berikut:
 - Pilih pengguna IAM untuk menambahkan pengguna IAM sebagai administrator portal Anda.
 - Pilih peran IAM untuk menambahkan peran IAM sebagai administrator portal Anda.
2. Pilih kotak centang untuk pengguna atau peran yang Anda inginkan sebagai administrator portal Anda. Ini menambahkan pengguna atau peran ke daftar administrator Portal.
3. Jika Anda menambahkan pengguna atau peran yang tidak Anda inginkan sebagai administrator, kosongkan kotak centang untuk pengguna atau peran tersebut.
4. Setelah selesai mengundang administrator portal, pilih Berikutnya.

Important

Pengguna atau peran harus memiliki `iotsitewise:DescribePortal` izin untuk masuk ke portal.

Note

Jika Anda menggunakan Pusat Identitas IAM sebagai penyimpanan identitas, dan Anda masuk ke akun AWS Organizations manajemen, Anda dapat memilih Buat pengguna untuk membuat pengguna Pusat Identitas IAM. IAM Identity Center mengirimkan email kepada pengguna baru agar mereka dapat mengatur kata sandi mereka. Anda kemudian dapat menetapkan pengguna ke portal sebagai administrator. Untuk informasi selengkapnya, lihat [Mengelola identitas di Pusat Identitas IAM](#).

Anda dapat mengubah daftar administrator portal nanti. Untuk informasi selengkapnya, lihat [Menambah atau menghapus administrator portal di AWS IoT SiteWise](#).

Note

Karena hanya administrator portal yang dapat membuat proyek dan menetapkan aset kepada mereka, Anda harus menentukan setidaknya satu administrator portal.

Sebagai langkah terakhir, Anda menambahkan pengguna yang dapat mengakses portal baru Anda.

Tambahkan pengguna portal di SiteWise Monitor

Anda mengontrol pengguna mana yang memiliki akses ke portal Anda. Di setiap portal, administrator portal membuat satu atau lebih proyek dan menetapkan pengguna portal sebagai pemilik atau pemirsa untuk setiap proyek. Setiap pemilik proyek dapat mengundang pengguna portal tambahan untuk memiliki atau melihat proyek.

Berdasarkan layanan otentikasi pengguna, pilih salah satu opsi berikut:

IAM Identity Center

Jika Anda ingin menambahkan pengguna ke daftar Pengguna, selesaikan langkah-langkah berikut.

Untuk menambahkan pengguna portal

1. Pilih pengguna dari daftar Pengguna untuk ditambahkan ke portal. Ini menambahkan pengguna ke daftar pengguna Portal. Jika Anda menggunakan SiteWise Monitor untuk pertama kalinya, Anda tidak perlu menambahkan administrator portal sebagai pengguna portal.

Note

Jika Anda menggunakan Pusat Identitas IAM sebagai penyimpanan identitas, dan Anda masuk ke akun AWS Organizations manajemen, Anda dapat memilih Buat pengguna untuk membuat pengguna Pusat Identitas IAM. IAM Identity Center mengirimkan email kepada pengguna baru agar mereka dapat mengatur kata sandi mereka. Anda kemudian dapat menetapkan pengguna ke portal sebagai pengguna. Untuk informasi selengkapnya, lihat [Mengelola identitas di Pusat Identitas IAM](#).

2. Jika Anda menambahkan pengguna yang tidak ingin Anda akses ke portal, kosongkan kotak centang untuk pengguna tersebut.
3. Setelah selesai memilih pengguna, pilih Tetapkan pengguna.

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 (2) Create user

Find resources

	Display name	Email
☐	Jane Doe	janedoe@example.com
☒	John Doe	johndoe@example.com

► Selected users (1)

Cancel Previous **Assign users**

IAM

Jika Anda melihat pengguna atau peran yang ingin Anda tambahkan dalam pengguna IAM atau daftar peran IAM, selesaikan langkah-langkah berikut.

Untuk menambahkan pengguna portal

1. Lakukan opsi berikut:

- Pilih pengguna IAM untuk menambahkan pengguna IAM sebagai pengguna portal.
- Pilih peran IAM untuk menambahkan peran IAM sebagai pengguna portal.

Jika Anda menggunakan SiteWise Monitor untuk pertama kalinya, Anda tidak perlu menambahkan administrator portal sebagai pengguna portal.

2. Pilih kotak centang untuk pengguna atau peran yang Anda inginkan sebagai pengguna portal. Ini menambahkan pengguna atau peran ke daftar pengguna Portal.
3. Jika Anda menambahkan pengguna yang tidak ingin Anda akses ke portal, kosongkan kotak centang untuk pengguna tersebut.
4. Setelah selesai memilih pengguna, pilih Tetapkan pengguna.

⚠ Important

Pengguna atau peran harus memiliki `iotsitewise:DescribePortal` izin untuk masuk ke portal.

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

☒	Name	Date created
☒	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](#)

☐

☐

☒

☐

☐

☐

☐

☐

☐

Name	Date created
[REDACTED]	
AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
AWSIoTSiteWiseMonitorServiceRole_GTnd004Wr	03-16-2021
AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
[REDACTED]	
[REDACTED]	
[REDACTED]	

► Portal users (2)

Remove

Cancel Previous **Assign users**

Selamat! Anda berhasil membuat portal, administrator portal yang ditetapkan, dan pengguna yang ditetapkan yang dapat menggunakan portal tersebut saat diundang untuk melakukannya. Administrator portal Anda sekarang dapat membuat proyek dan menambahkan aset ke proyek tersebut. Kemudian, pemilik proyek Anda dapat membuat dasbor untuk memvisualisasikan data untuk setiap aset proyek.

Anda dapat mengubah daftar pengguna portal nanti. Untuk informasi selengkapnya, lihat [Menambah atau menghapus pengguna portal di AWS IoT SiteWise](#).

Jika Anda perlu membuat perubahan pada portal, lihat [Mengelola portal SiteWise Monitor Anda](#).

Untuk memulai di portal, lihat [Memulai](#) di Panduan Aplikasi SiteWise Monitor.

Buat AWS IoT SiteWise dasbor ()AWS CLI

Saat Anda menentukan visualisasi (atau widget) di dasbor menggunakan AWS CLI, Anda harus menentukan informasi berikut dalam dokumen JSON. `dashboardDefinition` Definisi ini adalah parameter dari [CreateDashboard](#) dan [UpdateDashboard](#) operasi.

`widgets`

Daftar struktur definisi widget yang masing-masing berisi informasi berikut:

`type`

Jenis widget. AWS IoT SiteWise menyediakan jenis widget berikut:

- `sc-line-chart`— Bagan garis. Untuk informasi selengkapnya, lihat [Bagan garis](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- `sc-scatter-chart`— Bagan pencar. Untuk informasi selengkapnya, lihat [Bagan pencar](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- `sc-bar-chart`— Bagan batang. Untuk informasi selengkapnya, lihat [Bagan batang](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- `sc-status-grid`- Widget status yang menunjukkan nilai terbaru properti aset sebagai kisi. Untuk informasi selengkapnya, lihat [Widget status](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- `sc-status-timeline`— Widget status yang menunjukkan nilai historis properti aset sebagai garis waktu. Untuk informasi selengkapnya, lihat [Widget status](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- `sc-kpi`— Visualisasi indikator kinerja utama (KPI). Untuk informasi selengkapnya, lihat [widget KPI](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.
- `sc-table`— Sebuah widget tabel. Untuk informasi selengkapnya, lihat [Widget tabel](#) di Panduan AWS IoT SiteWise Monitor Aplikasi.

`title`

Judul widget.

x

Posisi horizontal widget, mulai dari kiri grid. Nilai ini mengacu pada posisi widget di grid dashboard.

y

Posisi vertikal widget, mulai dari bagian atas grid. Nilai ini mengacu pada posisi widget di grid dashboard.

width

Lebar widget, dinyatakan dalam jumlah spasi pada grid dashboard.

height

Ketinggian widget, dinyatakan dalam jumlah spasi pada grid dashboard.

metrics

Daftar struktur metrik yang masing-masing menentukan aliran data untuk widget ini. Setiap struktur dalam daftar harus berisi informasi berikut:

label

Label untuk ditampilkan untuk metrik ini.

type

Jenis sumber data untuk metrik ini. AWS IoT SiteWise menyediakan jenis metrik berikut:

- **iotsitewise**— Dasbor mengambil data untuk properti aset di AWS IoT SiteWise. Jika Anda memilih opsi ini, Anda harus menentukan `assetId` dan `propertyId` untuk metrik ini.

assetId

(Opsional) ID aset di AWS IoT SiteWise.

Bidang ini diperlukan jika Anda memilih **iotsitewise** untuk `type` dalam metrik ini.

propertyId

(Opsional) ID properti aset di AWS IoT SiteWise.

Bidang ini diperlukan jika Anda memilih **iotsitewise** untuk `type` dalam metrik ini.

analysis

(Opsional) Struktur yang mendefinisikan analisis, seperti garis tren, untuk ditampilkan untuk widget. Untuk informasi selengkapnya, lihat [Mengonfigurasi garis tren](#) di Panduan AWS IoT SiteWise Monitor Aplikasi. Anda dapat menambahkan salah satu dari setiap jenis garis tren per properti di widget. Struktur analisis berisi informasi berikut:

trends

(Opsional) Daftar struktur tren yang masing-masing menentukan analisis tren untuk widget ini. Setiap struktur dalam daftar berisi informasi berikut:

type

Jenis garis tren. Pilih opsi berikut:

- `linear-regression`— Menampilkan garis regresi linier. SiteWise Monitor menggunakan metode [kuadrat terkecil](#) untuk menghitung regresi linier.

annotations

(Opsional) Struktur anotasi yang mendefinisikan ambang batas untuk widget. Untuk informasi selengkapnya, lihat [Mengonfigurasi ambang batas dalam Panduan](#) Aplikasi. AWS IoT SiteWise Monitor Anda dapat menambahkan hingga enam anotasi per widget. Struktur anotasi berisi informasi berikut:

y

(Opsional) Daftar struktur anotasi yang masing-masing menentukan ambang horizontal untuk widget ini. Setiap struktur dalam daftar berisi informasi berikut:

comparisonOperator

Operator perbandingan untuk ambang batas. Pilih salah satu cara berikut:

- `LT`— Sorot properti yang memiliki setidaknya satu titik data kurang dari `value`.
- `GT`— Sorot properti yang memiliki setidaknya satu titik data lebih besar dari `value`.
- `LTE`— Sorot properti yang memiliki setidaknya satu titik data kurang dari atau sama dengan `value`.
- `GTE`— Sorot properti yang memiliki setidaknya satu titik data lebih besar dari atau sama dengan `value`.
- `EQ`— Sorot properti yang memiliki setidaknya satu titik data yang sama dengan `value`.

value

Nilai ambang batas untuk membandingkan titik data dengan `comparisonOperator`.

color

(Opsional) Kode heksadesimal 6 digit dari warna ambang batas. Visualisasi menampilkan legenda properti dalam warna ini untuk properti dengan setidaknya satu titik data yang memenuhi aturan ambang batas. Default ke black (). `#000000`

showValue

(Opsional) Apakah atau tidak untuk menunjukkan nilai ambang batas di margin widget. Default ke `true`.

properties

(Opsional) Kamus datar properti untuk widget. Anggota struktur ini bergantung pada konteks. AWS IoT SiteWise menyediakan widget berikut yang menggunakan `properties`:

- [Bagan garis](#), [bagan sebar](#), dan [bagan batang](#) memiliki properti berikut:

colorDataAcrossThresholds

(Opsional) Apakah atau tidak untuk mengubah warna data yang melintasi ambang batas dalam widget ini. Saat Anda mengaktifkan opsi ini, data yang melewati ambang batas muncul dalam warna yang Anda pilih. Default ke `true`.

- [Status grid](#) memiliki properti berikut:

labels

(Opsional) Struktur yang mendefinisikan label untuk ditampilkan pada grid status. Struktur label berisi informasi berikut:

showValue

(Opsional) Apakah akan menampilkan unit dan nilai untuk setiap properti aset di widget ini atau tidak. Default ke `true`.

Example Contoh definisi dasbor

Contoh berikut mendefinisikan dasbor dari payload yang disimpan dalam file JSON.

```
aws iotsitewise create-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \
```

```
--dashboard-name "Wind Farm Dashboard" \  
--dashboard-definition file://dashboard-definition.json
```

Contoh JSON berikut untuk `dashboard-definition.json` mendefinisikan dasbor dengan widget visualisasi berikut:

- Bagan garis yang memvisualisasikan total tenaga ladang angin di kiri atas dasbor. Bagan garis ini mencakup ambang batas yang menunjukkan kapan ladang angin menghasilkan daya yang lebih kecil daripada output minimum yang diharapkan. Grafik garis ini juga mencakup garis tren regresi linier.
- Diagram batang yang memvisualisasikan kecepatan angin untuk empat turbin di kanan atas dasbor.

Note

Contoh ini mewakili visualisasi diagram garis dan batang di dasbor. Dasbor ini mirip dengan [contoh dasbor ladang angin](#).

```
{  
  "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"  
  }  
]  
}  
]  
}
```

Nyalakan alarm untuk portal Anda AWS IoT SiteWise

Anda dapat mengaktifkan fitur alarm yang didukung oleh AWS IoT Events untuk portal Anda sehingga administrator portal dapat membuat, mengedit, dan menghapus model AWS IoT Events alarm di portal Monitor Anda SiteWise . Pemilik proyek dapat mengonfigurasi alarm. Pemirsa proyek dapat melihat detail alarm. Bagian ini menjelaskan bagaimana Anda dapat menggunakan AWS IoT SiteWise konsol untuk mengaktifkan fitur alarm untuk portal Anda.

Important

- Anda tidak dapat membuat alarm eksternal di portal Anda.
- Jika Anda ingin mengirim pemberitahuan alarm, Anda harus memilih Pusat Identitas IAM untuk layanan otentikasi pengguna.
- Fitur pemberitahuan alarm tidak tersedia di Tiongkok (Beijing Wilayah AWS).

Saat Anda mengonfigurasi dan membuat portal, Anda dapat mengaktifkan alarm dan pemberitahuan alarm di Langkah 2 Fitur tambahan. Berdasarkan layanan otentikasi pengguna, pilih salah satu opsi berikut:

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**

Untuk mengaktifkan alarm untuk portal

1. (Opsional) Pilih Aktifkan alarm.
 - Untuk peran AWS IoT SiteWise akses, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `iotevents:BatchPutMessage` izin dan hubungan kepercayaan yang memungkinkan `iot.amazonaws.com` dan `iotevents.amazonaws.com` untuk mengambil peran.
2. (Opsional) Pilih Aktifkan pemberitahuan alarm.
 - a. Untuk Pengirim, pilih pengirim.

⚠ Important

Anda harus memverifikasi alamat email pengirim di Amazon SES. Untuk informasi selengkapnya, lihat [Memverifikasi alamat email di Amazon SES](#), di Panduan Pengembang Layanan Email Sederhana Amazon.

- b. Untuk AWS Lambda peran, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `lambda:InvokeFunction` dan `sso-directory:DescribeUser` izin dan hubungan kepercayaan yang memungkinkan `iotevents.amazonaws.com` dan `lambda.amazonaws.com` untuk mengambil peran.
- c. Untuk AWS Lambda fungsi, pilih fungsi Lambda yang ada atau buat fungsi yang mengelola notifikasi alarm. Untuk informasi selengkapnya, lihat [Mengelola notifikasi alarm](#) di Panduan AWS IoT Events Pengembang.

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

i 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**

Untuk mengaktifkan alarm untuk portal

- (Opsional) Pilih Aktifkan alarm.

- Untuk peran AWS IoT SiteWise akses, gunakan peran yang ada atau buat peran dengan izin yang diperlukan. Peran ini membutuhkan `iotevents:BatchPutMessage` izin dan hubungan kepercayaan yang memungkinkan `iot.amazonaws.com` dan `iotevents.amazonaws.com` untuk mengambil peran.

Untuk informasi selengkapnya tentang alarm di SiteWise Monitor, lihat [Pemantauan dengan alarm](#) di Panduan AWS IoT SiteWise Aplikasi.

Mengaktifkan AWS IoT SiteWise portal Anda di tepi

Setelah Anda mengaktifkan portal Anda di tepi, portal ini tersedia di semua gateway SiteWise Edge dengan paket pemrosesan data diaktifkan di akun Anda.

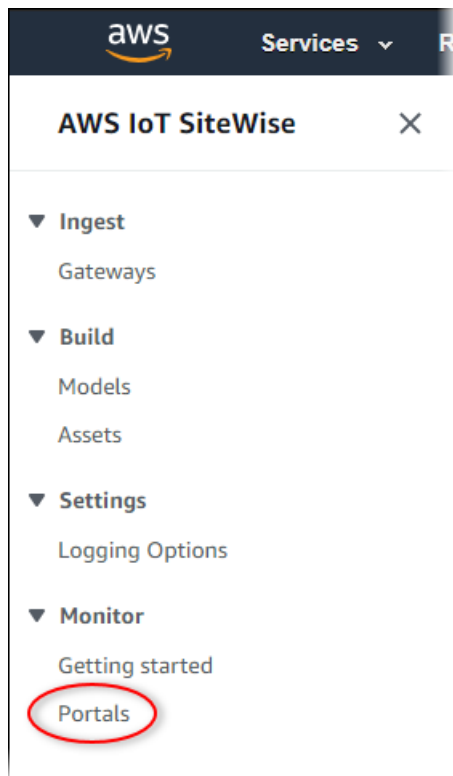
Untuk mengaktifkan portal di tepi

1. Di bagian konfigurasi Edge, nyalakan Aktifkan portal ini di tepi.
2. Pilih Buat.

Mengelola portal SiteWise Monitor Anda

Anda memiliki kemampuan untuk mengelola dan mengkonfigurasi berbagai aspek portal. Ini termasuk menambahkan dan menghapus pengguna atau administrator, mengatur izin dan peran pengguna, menyesuaikan URL portal, nama, menyetel informasi kontak dukungan, dan mengirim undangan email ke administrator portal.

1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Monitor, Portal.



3. Pilih portal, lalu pilih Lihat detail (atau pilih Nama portal).
4. Anda dapat melakukan salah satu tugas administratif berikut:
 - [Ubah detail portal di AWS IoT SiteWise](#)
 - [Menambah atau menghapus administrator portal di AWS IoT SiteWise](#)
 - [Kirim undangan email ke administrator portal](#)
 - [Menambah atau menghapus pengguna portal di AWS IoT SiteWise](#)
 - [Hapus portal di AWS IoT SiteWise](#)

Untuk informasi tentang cara membuat portal, lihat [Memulai dengan AWS IoT SiteWise Monitor \(Klasik\)](#).

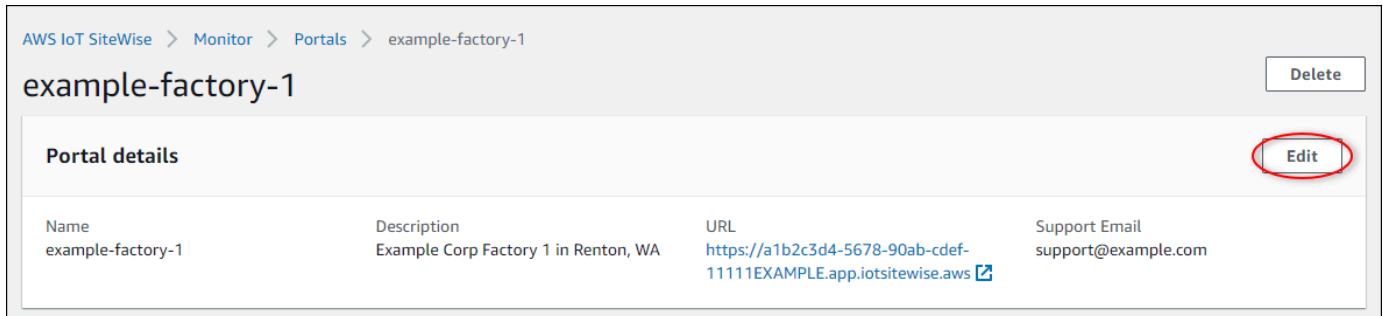
Topik

- [Ubah detail portal di AWS IoT SiteWise](#)
- [Menambah atau menghapus administrator portal di AWS IoT SiteWise](#)
- [Kirim undangan email ke administrator portal](#)
- [Menambah atau menghapus pengguna portal di AWS IoT SiteWise](#)
- [Hapus portal di AWS IoT SiteWise](#)

Ubah detail portal di AWS IoT SiteWise

Anda dapat mengubah nama portal, deskripsi, branding, email dukungan, dan izin.

1. Pada halaman detail portal, di bagian Detail portal, pilih Edit.

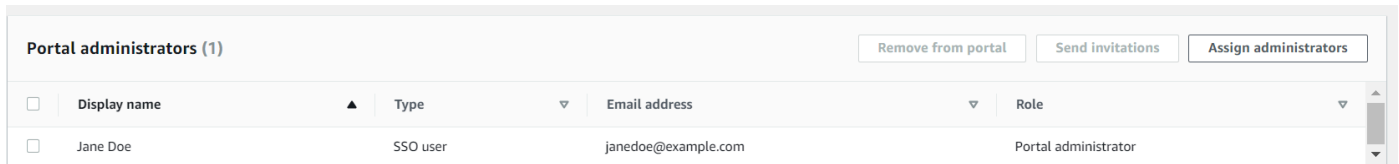


2. Perbarui Nama, Deskripsi, pencitraan merek Portal, email kontak Dukungan, atau Izin.
3. Setelah selesai, pilih Simpan.

Menambah atau menghapus administrator portal di AWS IoT SiteWise

Dalam beberapa langkah, Anda dapat menambah atau menghapus pengguna sebagai administrator untuk portal. Berdasarkan layanan otentikasi pengguna, pilih salah satu opsi berikut.

IAM Identity Center



Untuk menambahkan administrator portal

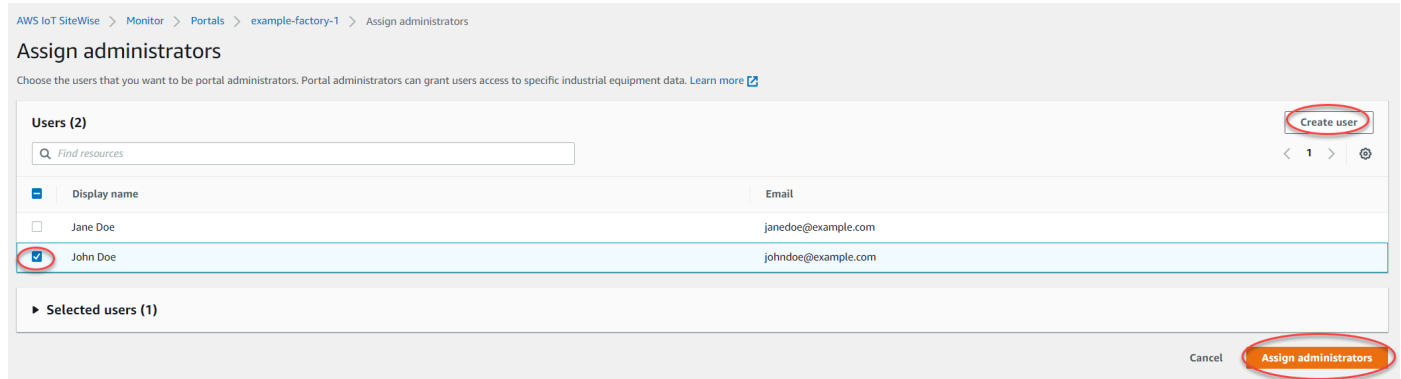
1. Pada halaman detail portal, di bagian Administrator portal, pilih Tetapkan administrator.
2. Pada halaman Tetapkan administrator, pilih kotak centang untuk ditambahkan pengguna ke portal sebagai administrator.

Note

Jika Anda menggunakan Pusat Identitas IAM sebagai penyimpanan identitas, dan Anda masuk ke akun AWS Organizations manajemen, Anda dapat memilih Buat pengguna untuk membuat pengguna Pusat Identitas IAM. IAM Identity Center mengirimkan email kepada pengguna baru untuk mengatur kata sandi mereka.

Anda kemudian dapat menetapkan pengguna ke portal sebagai administrator. Untuk informasi selengkapnya, lihat [Mengelola identitas di Pusat Identitas IAM](#).

3. Pilih Tetapkan administrator.



Untuk menghapus administrator portal

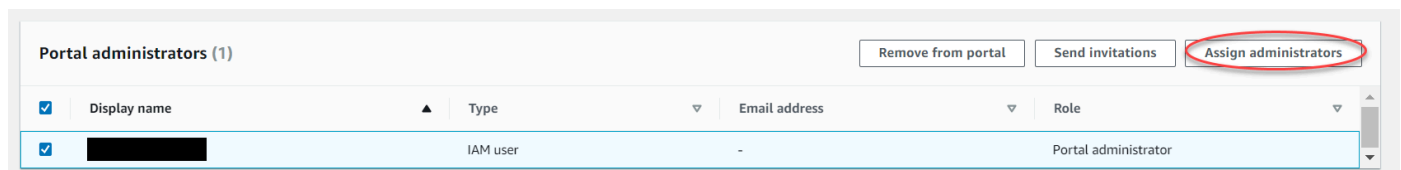
- Pada halaman detail portal, di bagian Administrator portal, pilih kotak centang untuk dihapus setiap pengguna, lalu pilih Hapus dari portal.



Note

Kami menyarankan Anda memilih setidaknya satu administrator portal.

IAM



Untuk menambahkan administrator portal

1. Pada halaman detail portal, di bagian Administrator portal, pilih Tetapkan administrator.
2. Pada halaman Tetapkan administrator, lakukan hal berikut:
 - Pilih pengguna IAM, jika Anda ingin menambahkan pengguna IAM sebagai administrator portal Anda.

- Pilih peran IAM, jika Anda ingin menambahkan peran IAM sebagai administrator portal Anda.
3. Pilih kotak centang untuk pengguna atau peran yang Anda inginkan sebagai administrator portal Anda. Ini menambahkan pengguna atau peran ke daftar administrator Portal.
 4. Pilih Tetapkan administrator.

Important

Pengguna atau peran harus memiliki `iotsitewise:DescribePortal` izin untuk masuk ke portal.

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](#)

☒	Name	Date created
☒	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 `lotsitewise:DescribePortal` permission to sign in to the portal.Users **Roles**

IAM roles (66)

[Manage roles in IAM console](#)

Find role name

< 1 2 3 4 5 6 7 >

☐	Name	Date created
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

Portal administrators (2)

Remove

Cancel **Assign administrators**

Untuk menghapus administrator portal

- Pada halaman detail portal, di bagian Administrator portal, pilih kotak centang untuk dihapus setiap pengguna, lalu pilih Hapus dari portal.

Note

Meninggalkan portal tanpa administrator portal tidak disarankan.

Kirim undangan email ke administrator portal

Anda dapat mengirim undangan email ke administrator portal.

- Pada halaman detail portal, di bagian Administrator portal, pilih kotak centang untuk administrator portal.

Portal administrators (1)				Remove from portal	Send invitations	Assign users
☒	Display name	Email address	Role			
☒	John Doe	john.doe@example.com	Portal administrator			

- Pilih Kirim undangan. Klien email Anda terbuka, dan undangan diisi di badan pesan.

Anda dapat menyesuaikan email sebelum mengirimkannya ke administrator portal Anda.

Menambah atau menghapus pengguna portal di AWS IoT SiteWise

Anda memilih pengguna mana yang memiliki akses ke portal Anda. Pengguna portal muncul dalam daftar pengguna dalam portal SiteWise Monitor. Dari daftar ini, administrator portal dapat menambahkan pemilik proyek, dan pemilik proyek dapat menambahkan pemirsa proyek.

Note

Administrator portal dan pengguna portal Anda mungkin menghubungi Anda melalui email dukungan portal jika mereka membutuhkan Anda untuk menambah atau menghapus pengguna.

Berdasarkan layanan otentikasi pengguna, pilih salah satu opsi berikut.

IAM Identity Center

Portal users (1)						Remove from portal	Assign users
☐	Display name	Type	Email address	Role			
☐	John Doe	SSO user	johndoe@example.com	Portal viewer			

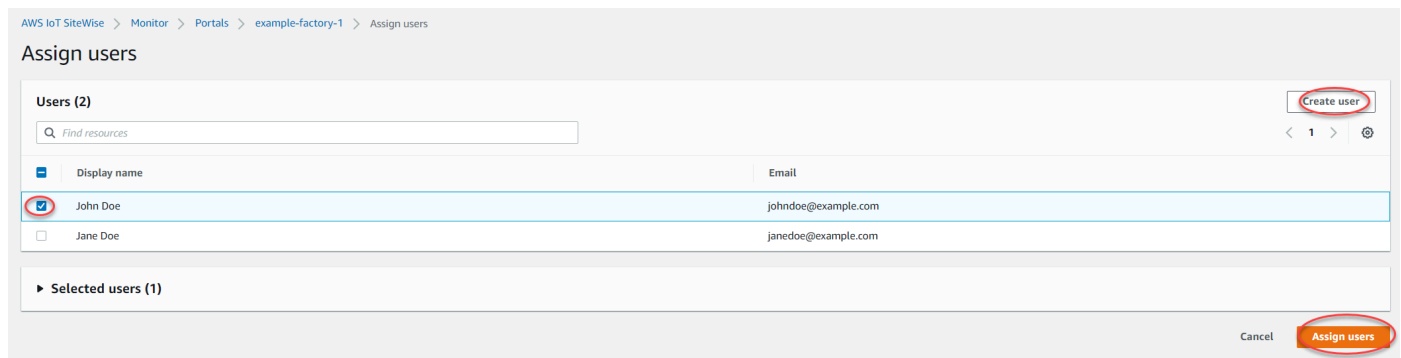
Untuk menambahkan pengguna portal

1. Pada halaman detail portal, di bagian Pengguna portal, pilih Tetapkan pengguna.
2. Pada halaman Tetapkan pengguna, pilih kotak centang untuk ditambahkan pengguna ke portal.

Note

Jika Anda menggunakan Pusat Identitas IAM sebagai penyimpanan identitas, dan Anda masuk ke akun AWS Organizations manajemen, Anda dapat memilih Buat pengguna untuk membuat pengguna Pusat Identitas IAM. IAM Identity Center mengirimkan email kepada pengguna baru untuk mengatur kata sandi mereka. Anda kemudian dapat menetapkan pengguna ke portal sebagai pengguna. Untuk informasi selengkapnya, lihat [Mengelola identitas di Pusat Identitas IAM](#).

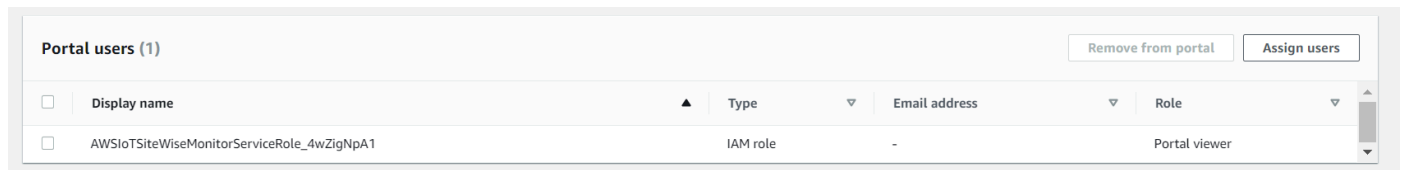
3. Pilih Tetapkan pengguna.



Untuk menghapus pengguna portal

- Pada halaman detail portal, di bagian Pengguna portal, pilih kotak centang untuk dihapus pengguna dari portal, lalu pilih Hapus dari portal.

IAM



Untuk menambahkan pengguna portal

1. Pada halaman detail portal, di bagian Pengguna portal, pilih Tetapkan pengguna.
2. Pada halaman Tetapkan pengguna, lakukan hal berikut:
 - Pilih pengguna IAM untuk menambahkan pengguna IAM sebagai pengguna portal Anda.
 - Pilih peran IAM untuk menambahkan peran IAM sebagai pengguna portal Anda.
3. Pilih kotak centang untuk pengguna atau peran yang ingin Anda tambahkan sebagai pengguna portal Anda. Ini menambahkan pengguna atau peran ke daftar pengguna Portal.
4. Pilih Tetapkan pengguna.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

☒	Name	Date created
☒	[REDACTED]	11-08-2019

► **Portal users (1)** [Remove](#)

Cancel [Assign users](#)

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

☐	Name	Date created
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_RHINLNC5-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

► **Portal users (2)** [Remove](#)

Cancel [Assign users](#)

Untuk menghapus pengguna portal

- Pada halaman detail portal, di bagian Pengguna portal, pilih kotak centang untuk dihapus pengguna dari portal, lalu pilih Hapus dari portal.

Important

Pengguna atau peran harus memiliki `iotsitewise:DescribePortal` izin untuk masuk ke portal.

Hapus portal di AWS IoT SiteWise

Anda dapat menghapus portal jika Anda membuatnya untuk tujuan pengujian atau jika Anda membuat duplikat portal yang sudah ada.

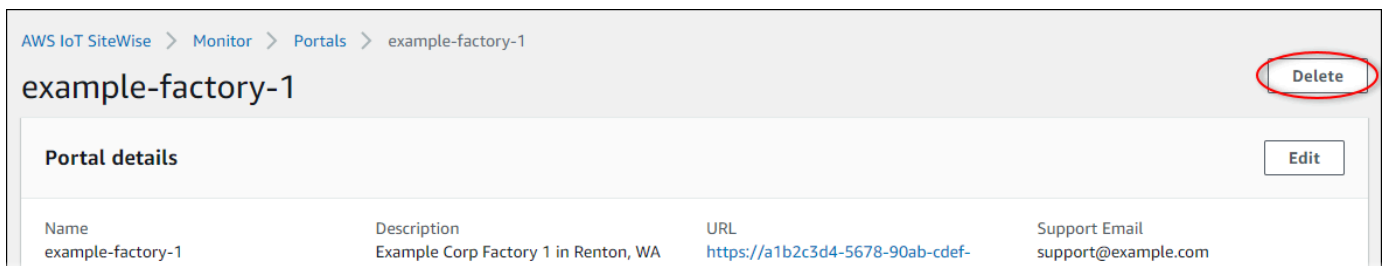
Note

Anda harus terlebih dahulu menghapus semua dasbor dan proyek secara manual di portal sebelum Anda dapat menghapus portal. Untuk informasi selengkapnya, lihat [Menghapus proyek](#) dan [Menghapus dasbor di Panduan](#) Aplikasi SiteWise Monitor.

1. Pada halaman detail portal, pilih Hapus.

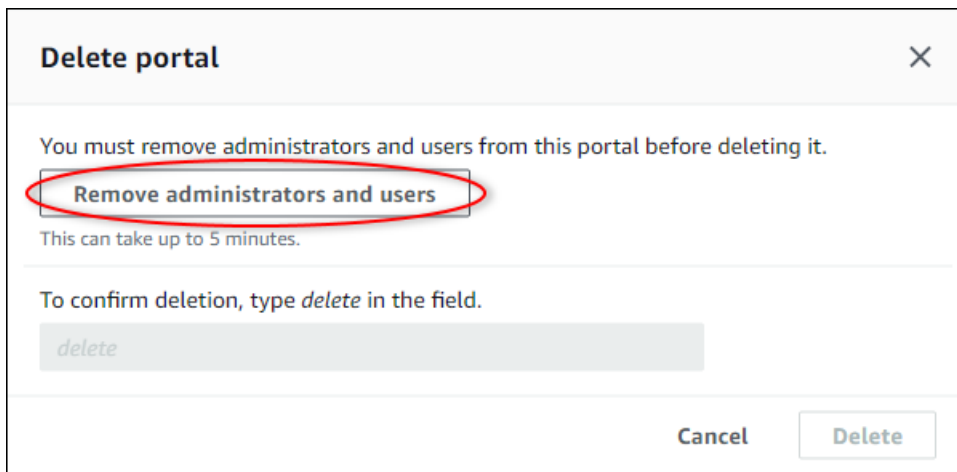
Important

Ketika Anda menghapus portal, Anda kehilangan semua proyek yang berisi portal, dan semua dasbor di setiap proyek. Tindakan ini tidak dapat dibatalkan. Data aset Anda tidak terpengaruh.

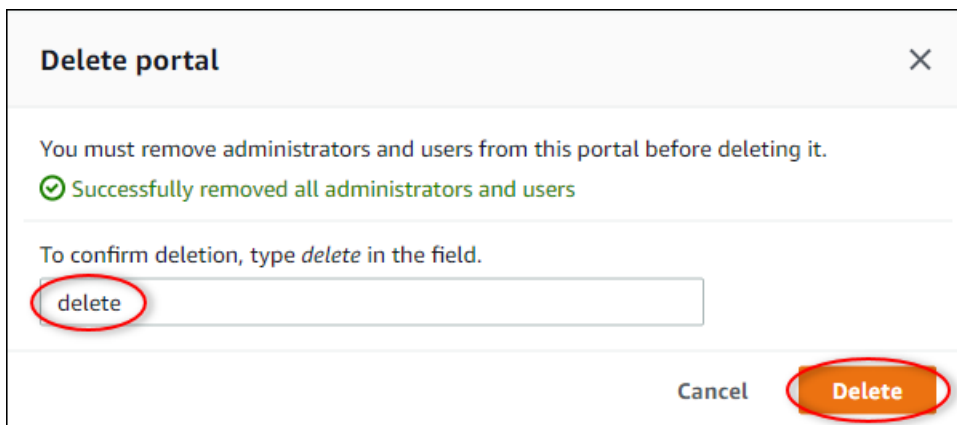


2. Di kotak dialog Hapus portal, pilih Hapus admin dan pengguna.

Anda harus menghapus administrator dan pengguna dari portal sebelum Anda dapat menghapusnya. Jika portal Anda tidak memiliki administrator atau pengguna, tombol tidak muncul, dan Anda dapat melompat ke langkah berikutnya.



3. Jika Anda yakin ingin menghapus seluruh portal, masukkan **delete** di bidang untuk mengonfirmasi penghapusan.



4. Pilih Hapus.

Memulai dengan AWS IoT SiteWise Monitor (AI-aware) - pratinjau

Sebagai AWS administrator untuk organisasi Anda, Anda dapat membuat portal dari AWS IoT SiteWise konsol, memungkinkan anggota organisasi Anda untuk melihat data Anda AWS IoT SiteWise . Selesaikan langkah-langkah berikut untuk memulai.

1. Konfigurasi dan buat portal.
2. Tambahkan administrator portal dan kirim email undangan.
3. Tambahkan pengguna portal.

Setelah Anda membuat portal, administrator portal dapat membuat proyek dan menambahkan pengguna ke proyek. Anggota proyek kemudian membuat dasbor untuk memvisualisasikan data

yang terhubung AWS IoT SiteWise, memungkinkan mereka memantau kinerja perangkat, proses, dan peralatan mereka yang terhubung.

 Note

Saat menambahkan pengguna atau administrator ke portal, hindari membuat kebijakan AWS Identity and Access Management (IAM) yang membatasi izin pengguna, seperti IP terbatas. Kebijakan terlampir dengan izin terbatas tidak akan dapat terhubung ke AWS IoT SiteWise portal.

Buat proyek untuk dibagikan dengan tim Anda. Pemilik proyek kemudian dapat membuat dasbor untuk memvisualisasikan properti aset yang membantu pemirsa proyek memahami kinerja perangkat, proses, dan peralatan. Ini juga memberikan pandangan operasi yang konsisten untuk tim Anda.

Dasbor membantu memvisualisasikan dan memahami data proyek Anda. Ini membantu bisnis dan pengguna aplikasi melacak AWS IoT perangkat dan data mereka. Pilih jenis visualisasi yang paling baik menampilkan data Anda untuk kebutuhan Anda. Atur ulang dan ukur ulang visualisasi untuk membuat tata letak yang sesuai dengan tim Anda. Jelajahi aset dan data perangkat, proses, dan peralatan Anda, serta identifikasi masalah dengan cepat dan tingkatkan efisiensi operasional.

Topik

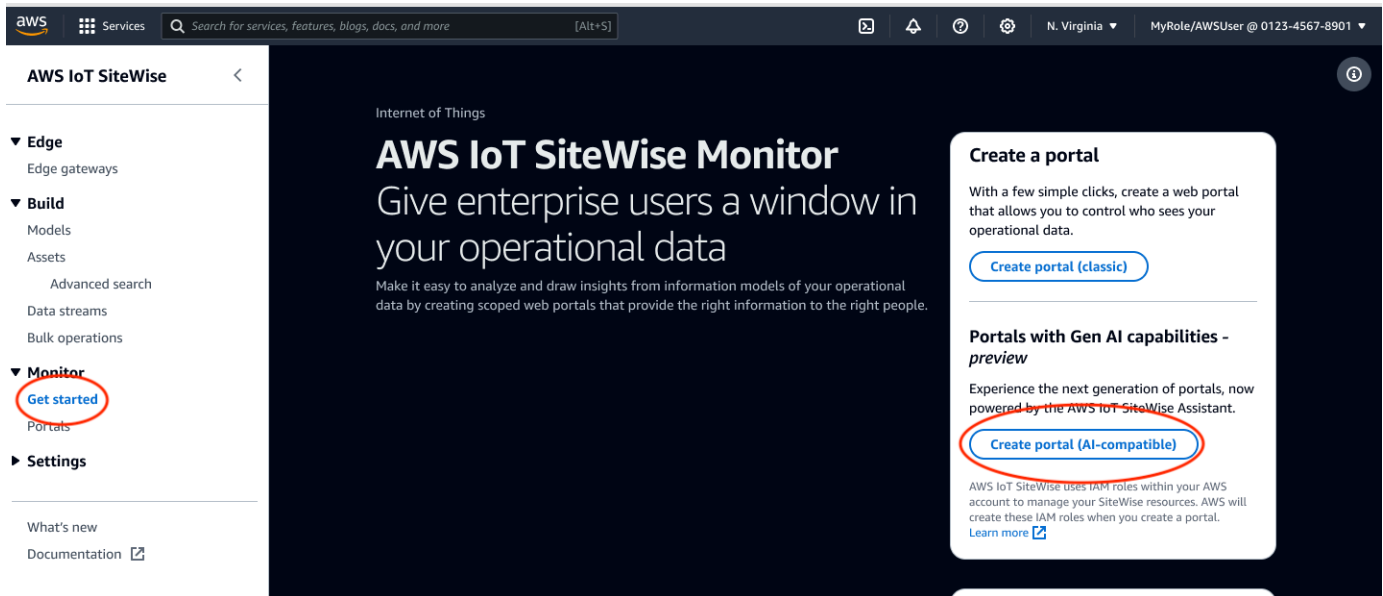
- [Buat portal](#)
- [Konfigurasi portal Anda](#)
- [Kelola portal Anda](#)
- [Hapus portal](#)
- [Buat dasbor dengan AWS CLI](#)
- [Login portal](#)
- [Membuat proyek](#)
- [Perbarui proyek](#)
- [Hapus proyek](#)
- [Membuat sebuah Dasbor](#)
- [Perbarui dasbor](#)
- [Hapus dasbor](#)
- [Konfigurasi dasbor](#)

Buat portal

Anda membuat portal SiteWise Monitor di AWS IoT SiteWise konsol.

Untuk membuat portal

1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Monitor, Memulai.
3. Pilih Buat portal (Ai-aware).



Selanjutnya, Anda harus memberikan beberapa informasi dasar untuk mengkonfigurasi portal Anda.

Konfigurasi portal Anda

Pengguna Anda menggunakan portal untuk melihat data Anda. Anda dapat menyesuaikan nama portal, deskripsi, branding, otentikasi pengguna, email kontak dukungan, dan izin.

Langkah-langkah untuk mengkonfigurasi portal:

1. Masukkan nama untuk portal Anda.
2. (Opsional) Masukkan deskripsi untuk portal Anda. Jika Anda memiliki beberapa portal, gunakan deskripsi yang bermakna untuk membantu Anda melacak isi setiap portal.

3. (Opsional) Unggah gambar untuk menampilkan merek Anda di portal. Pilih persegi, gambar PNG. Jika Anda mengunggah gambar non-persegi, portal menskalakan gambar menjadi persegi.
4. Masukkan alamat email di kotak email kontak Support untuk masalah dukungan.
5. Di kotak Autentikasi pengguna, pilih opsi berikut:
 - Pilih Pusat Identitas IAM jika pengguna portal Anda masuk ke portal ini dengan nama pengguna dan kata sandi perusahaan mereka.

Jika Anda belum mengaktifkan Pusat Identitas IAM di akun Anda, lakukan hal berikut:

- a. Pilih Create user (Buat pengguna).
- b. Pada halaman Buat pengguna, untuk membuat portal pertama, masukkan alamat email pengguna, nama depan, dan nama belakang, lalu pilih Buat pengguna.

 Note

Support untuk kredensi IAM akan segera hadir.

6. Pilih salah satu opsi berikut di bagian Akses layanan:
 - Pilih Buat dan gunakan peran layanan baru. Secara default, SiteWise Monitor secara otomatis membuat peran layanan untuk setiap portal. Peran ini memungkinkan pengguna portal Anda mengakses AWS IoT SiteWise sumber daya Anda. Untuk informasi selengkapnya, lihat [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#).
 - Pilih Gunakan peran layanan yang ada, lalu pilih peran target.
7. Pilih untuk mengaktifkan AWS IoT SiteWise Asisten untuk portal ini. AWS IoT SiteWise Asisten menyediakan analisis data yang cepat, wawasan waktu nyata, dan rekomendasi terpandu.

 Note

Mengaktifkan AWS IoT SiteWise Asisten akan dikenakan biaya. Untuk menggunakan solusi dan panduan pengetahuan tingkat perusahaan, Anda harus memiliki kumpulan data yang terkait dengan indeks Amazon Kendra.

8. (Opsional) Tambahkan tag untuk portal Anda. Untuk informasi selengkapnya, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).
9. Pilih Buat portal. AWS IoT SiteWise akan membuat portal Anda.

Note

Jika Anda menutup konsol, Anda dapat menyelesaikan proses penyiapan dengan menambahkan administrator dan pengguna. Untuk informasi selengkapnya, lihat [Menambah atau menghapus administrator portal](#). Jika Anda tidak ingin menyimpan portal ini, hapus agar tidak menggunakan sumber daya. Untuk informasi selengkapnya, lihat [Hapus portal](#).

Pesan muncul saat portal Anda dibuat.

Successfully created portal "example portal".

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > example portal

example portal [Edit](#) [Delete](#) [Open portal](#)

Portal details

Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

Setelah portal dibuat, itu tercantum di bagian Portal. Bagian detail Portal mencantumkan nama, deskripsi, ID, URL, status, tanggal terakhir diperbarui dan dibuat, branding portal, dan email dukungan untuk setiap portal.

Kolom Status dapat menjadi salah satu nilai berikut.

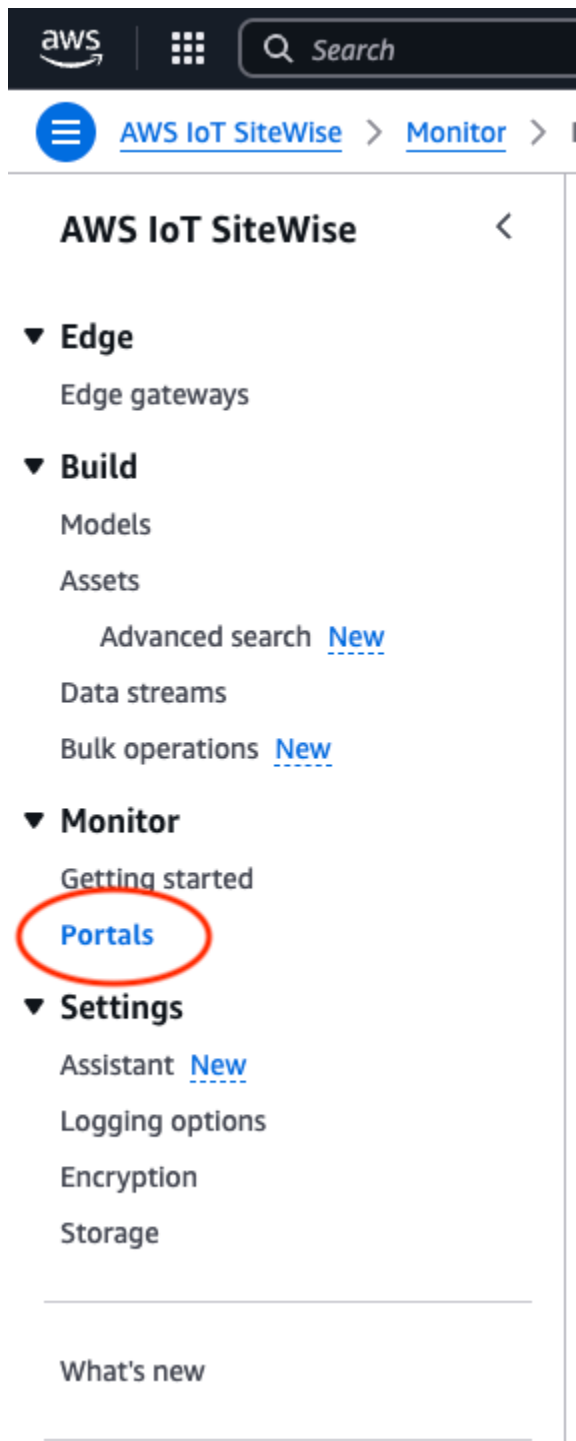
- **AWS IoT SiteWise CREATING** - sedang memproses permintaan Anda untuk membuat portal. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- **UPDATE** - AWS IoT SiteWise sedang memproses permintaan Anda untuk memperbarui portal. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- **PENDING** - AWS IoT SiteWise sedang menunggu propagasi catatan DNS selesai. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya. Anda dapat menghapus portal saat statusnya TERTUNDA.

- **MENGHAPUS** - AWS IoT SiteWise adalah memproses permintaan Anda untuk menghapus portal. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- **AKTIF** - Ketika portal menjadi aktif, pengguna portal Anda dapat mengaksesnya.
- **GAGAL** - AWS IoT SiteWise tidak dapat memproses permintaan Anda untuk membuat, memperbarui, atau menghapus portal. Jika Anda mengaktifkan AWS IoT SiteWise untuk mengirim log ke Amazon CloudWatch Logs, Anda dapat menggunakan log ini untuk memecahkan masalah. Untuk informasi selengkapnya, lihat [Pemantauan AWS IoT SiteWise dengan CloudWatch Log](#).

Kelola portal Anda

Anda memiliki kemampuan untuk mengelola dan mengkonfigurasi berbagai aspek portal. Ini termasuk menambahkan dan menghapus administrator, mengatur izin dan peran, menyesuaikan nama, deskripsi, menyiapkan email dukungan, dan mengundang administrator portal.

1. Masuk ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Monitor, Portal.



3. Pilih portal, lalu pilih Buka portal (atau pilih Nama portal).
4. Anda dapat melakukan salah satu tugas administratif berikut:
 - [Edit atribut portal](#)
 - [Menambah atau menghapus administrator portal](#)

- [Kirim undangan email ke administrator portal](#)
- [Hapus portal di AWS IoT SiteWise](#)

Edit atribut portal

Anda dapat mengubah nama portal, deskripsi, branding, email dukungan, dan akses layanan.

1. Pada halaman detail portal, di bagian Detail portal, pilih Edit.

Successfully created portal "example portal".

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > example portal

example portal Edit Delete Open portal

Portal details		
Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

2. Perbarui Nama, Deskripsi, branding Portal, Email kontak Dukungan, Akses AWS IoT SiteWise Asisten atau Layanan.
3. Setelah Anda selesai, pilih Simpan perubahan.

Menambah atau menghapus administrator portal

Dalam beberapa langkah, Anda dapat menambah atau menghapus pengguna sebagai administrator untuk portal. Berdasarkan layanan otentikasi pengguna, pilih salah satu opsi berikut.

IAM Identity Center

Administrators (2) [Info](#) Remove from portal Send invitations Assign administrators

☐	Display name	Email address
☐	Jane Doe	janedoe@amazon.com
☐	John Doe	johndoe@amazon.com

Untuk menambahkan administrator portal

1. Pada halaman detail portal, di bagian Administrator, pilih Tetapkan administrator.
2. Pada halaman Tetapkan administrator, pilih pengguna yang akan ditambahkan ke portal sebagai administrator.

Note

Jika Anda menggunakan Pusat Identitas IAM sebagai penyimpanan identitas, dan Anda masuk ke akun AWS Organizations manajemen, Anda dapat memilih Buat pengguna untuk membuat pengguna Pusat Identitas IAM. IAM Identity Center mengirimkan email kepada pengguna baru untuk mengatur kata sandi mereka. Anda kemudian dapat menetapkan pengguna ke portal sebagai administrator. Untuk informasi selengkapnya, lihat [Mengelola identitas di Pusat Identitas IAM](#).

3. Pilih Tetapkan administrator.

Assign administrators [Info](#)

Choose portal administrators from the users list. Portal administrators grant access to specific industrial equipment data. [Learn more](#)

Users (11) Create user

X 2 matches < 1 > ⚙

☐	Name	Email
☒	johndoe@amazon.com	johndoe@amazon.com
☐	janedoe@amazon.com	janedoe@amazon.com

Cancel Assign administrators

Untuk menghapus administrator portal

- Pada halaman detail portal, di bagian Administrator portal, pilih kotak centang untuk dihapus setiap pengguna, lalu pilih Hapus dari portal.

Note

Administrator (#) mencantumkan jumlah administrator untuk portal. Anda dapat menambahkan beberapa administrator portal untuk mengelola dan mengerjakan proyek.

Kirim undangan email ke administrator portal

Anda dapat mengirim undangan email ke administrator portal.

1. Pada halaman detail portal, di bagian Administrator, pilih kotak centang untuk administrator portal.
2. Pilih Kirim undangan. Klien email Anda terbuka, dan undangan diisi di badan pesan.

Anda dapat menyesuaikan email sebelum mengirimkannya ke administrator portal Anda.

Hapus portal

Anda dapat menghapus portal jika Anda membuatnya untuk tujuan pengujian atau jika Anda membuat duplikat portal yang sudah ada.

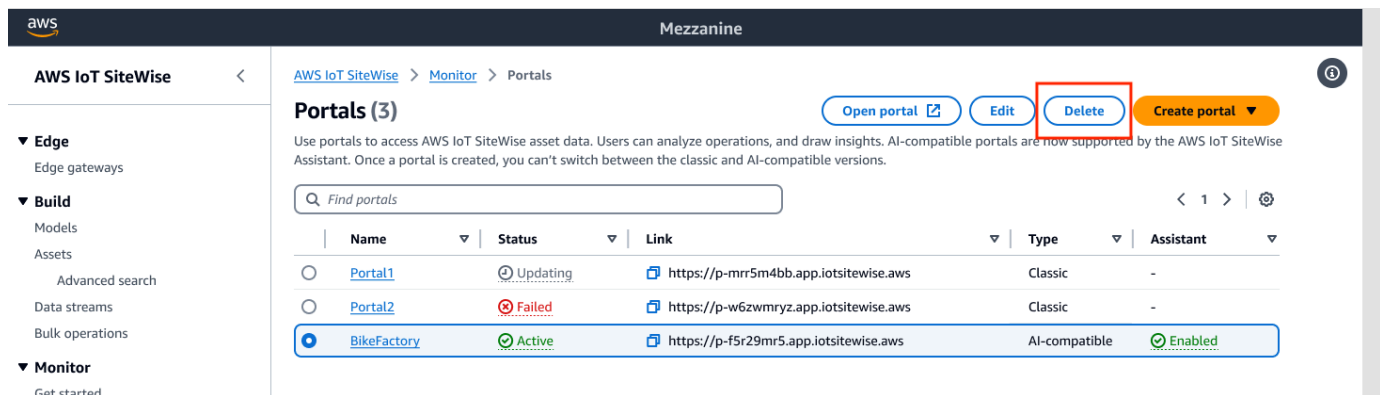
Note

Anda harus terlebih dahulu menghapus semua dasbor dan proyek secara manual di portal sebelum Anda dapat menghapus portal.

1. Pada halaman detail portal, pilih Hapus.

Important

Ketika Anda menghapus portal, Anda kehilangan semua proyek yang berisi portal, dan semua dasbor di setiap proyek. Tindakan ini tidak dapat dibatalkan. Data aset Anda tidak terpengaruh.



2. Di kotak dialog Hapus portal, pilih Hapus admin dan pengguna.

Anda harus menghapus administrator dan pengguna dari portal sebelum Anda dapat menghapusnya. Jika portal Anda tidak memiliki administrator atau pengguna, tombol tidak muncul, dan Anda dapat melompat ke langkah berikutnya.

3. Jika Anda yakin ingin menghapus seluruh portal, masukkan **confirm** di bidang untuk mengonfirmasi penghapusan.
4. Pilih Hapus.

Buat dasbor dengan AWS CLI

Saat Anda menentukan visualisasi (atau widget) di dasbor menggunakan AWS CLI, Anda harus menentukan informasi berikut dalam dokumen JSON. `dashboardDefinition` Definisi ini adalah parameter dari [CreateDashboard](#) dan [UpdateDashboard](#) operasi.

displaySettings

Pengaturan tampilan dengan parameter berikut:

- `numRows`— Jumlah baris dalam tata letak dasbor. Setiap baris lebar `CellSize`.
- `numColumns`— Jumlah kolom dalam tata letak dasbor. Setiap kolom lebar `CellSize`.
- `cellSize`— (Opsional) Ukuran sel dalam tata letak dalam piksel. Itu harus angka positif. Defaultnya adalah 10.
- `significantDigits`— (Opsional) Jumlah digit signifikansi untuk ditampilkan di dasbor. Defaultnya adalah 4.

querySettings

Informasi query dengan parameter berikut:

- **refreshRate**— (Opsional) Tingkat di mana data menyegarkan dalam milidetik. Menerima nilai-nilai berikut - 1000, 5000, 10000, 60000, 300000.

defaultViewport

Jika tidak disediakan, default ke lima menit terakhir. Berisi parameter berikut:

- **duration**— (Opsional) Menentukan seberapa jauh ke masa lalu untuk menanyakan data mulai dari saat ini.
- **start**— (Opsional) Ini adalah tipe Tanggal. Rentang waktu mulai untuk kueri data. Membutuhkan end tanggal yang ditentukan.
- **end**— (Opsional) Ini adalah tipe Tanggal. Rentang waktu akhir untuk kueri data. Membutuhkan start tanggal yang ditentukan.

widgets

Daftar struktur definisi widget yang berisi informasi berikut:

type

Jenis widget. AWS IoT SiteWise menyediakan jenis widget berikut:

- **xy-plot**— Bagan garis atau plot pencar tergantung pada konfigurasi.
- **bar-chart**— Bagan batang.
- **kpi-chart**— Grafik indikator kinerja utama.
- **status-timeline**— Widget status yang memvisualisasikan dan menavigasi data deret waktu dari satu atau lebih sumber data.
- **text**- Widget teks.
- **table**- Widget tabel.

id

Pengidentifikasi unik untuk widget.

x

Posisi horizontal widget, mulai dari kiri dasbor. Nilai ini mengacu pada posisi widget di grid dashboard.

y

Posisi vertikal widget, mulai dari bagian atas dasbor. Nilai ini mengacu pada posisi widget di grid dashboard.

z

Urutan relatif widget. Widget nilai Z yang lebih besar ditampilkan di depan widget nilai Z yang lebih rendah, jika tumpang tindih.

width

Lebar widget, dinyatakan dalam jumlah sel di dasbor.

height

Ketinggian widget, dinyatakan dalam jumlah sel di dasbor.

properties

Daftar properti widget. Ini bervariasi menurut jenis widget. Lihat [Kit Aplikasi IoT](#) untuk detailnya.

Example Contoh definisi dasbor

Contoh berikut mendefinisikan dasbor dari payload yang disimpan dalam file JSON.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --dashboard-name "Example Dashboard" \  
  --dashboard-definition file://dashboard-definition.json
```

Contoh JSON berikut untuk `dashboard-definition.json` mendefinisikan dasbor dengan widget visualisasi berikut:

```
{  
  "displaySettings": {  
    "numColumns": 200,  
    "numRows": 1000,
```

```

    "cellSize": 20,
    "significantDigits": 4
  },
  "widgets": [{
    "id": "0t73JcxUoc6oEXAMPLE",
    "type": "xy-plot",
    "width": 33,
    "height": 20,
    "x": 0,
    "y": 0,
    "z": 0,
    "properties": {
      "aggregationType": "AVERAGE",
      "queryConfig": {
        "source": "iotsitewise",
        "query": {
          "assets": [{
            "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
            "properties": [{
              "propertyId": "97cc61f4-57a4-4c5f-a82c-EXAMPLE",
              "refId": "692ce941-f3d9-4074-a297-EXAMPLE",
              "aggregationType": "AVERAGE",
              "color": "#7d2105",
              "resolution": "1m"
            }]
          }]
        },
        "properties": [],
        "assetModels": [],
        "alarms": [],
        "alarmModels": []
      }
    },
    "line": {
      "connectionStyle": "linear",
      "style": "solid"
    },
    "symbol": {
      "style": "filled-circle"
    },
    "axis": {
      "yVisible": true,
      "xVisible": true
    },
    "legend": {

```

```

        "visible": true,
        "position": "right",
        "width": "30%",
        "height": "30%",
        "visibleContent": {
            "unit": true,
            "asset": true,
            "latestValue": true,
            "latestAlarmStateValue": true,
            "maxValue": false,
            "minValue": false
        }
    }
}
}, {
    "id": "fto7rF40Ny1EXAMPLE-G",
    "type": "bar-chart",
    "width": 33,
    "height": 20,
    "x": 0,
    "y": 20,
    "z": 0,
    "properties": {
        "aggregationType": "AVERAGE",
        "queryConfig": {
            "source": "iotsitewise",
            "query": {
                "assets": [{
                    "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
                    "properties": [{
                        "propertyId": "c84ca8f3-3dea-478a-afec-EXAMPLE",
                        "aggregationType": "AVERAGE",
                        "refId": "2960b958-2034-4d6e-bcc2-EXAMPLE"
                    }]
                }]
            },
            "properties": [],
            "assetModels": [],
            "alarms": [],
            "alarmModels": [],
            "requestSettings": {
                "aggregation": "AVERAGE"
            }
        }
    }
},

```

```

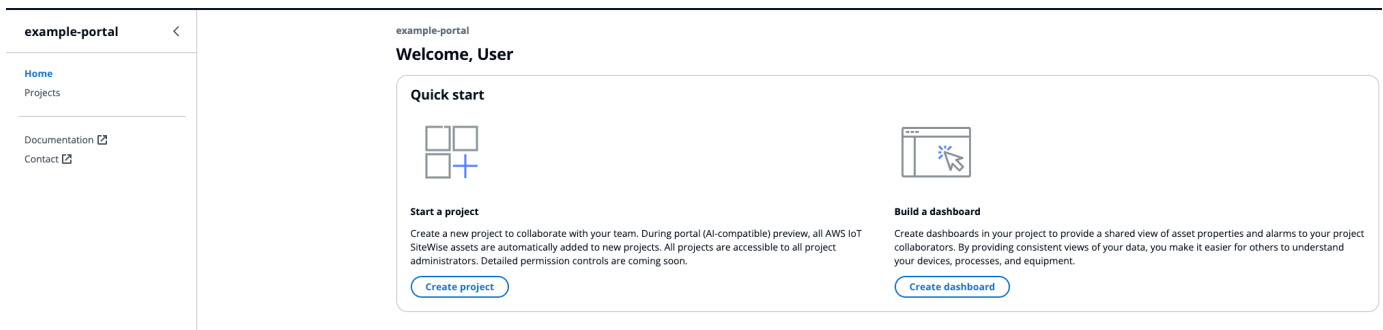
    "axis": {
      "showX": true,
      "showY": true
    },
    "styleSettings": {
      "2960b958-2034-4d6e-bcc2-360f1f02e505": {
        "color": "#7d2105"
      }
    }
  }
},
"querySettings": {
  "refreshRate": 5000
}
}

```

Login portal

Penggunaan Login

1. Di browser Anda, masukkan URL aplikasi.
2. Masukkan nama pengguna dan kata sandi Anda dan klik tombol Masuk.
3. Anda sekarang masuk ke aplikasi.



Membuat proyek

Buat proyek

1. Sebuah proyek dibuat dengan dua cara:
 - a. Di halaman Beranda, di bagian Selamat Datang di bawah Mulai cepat, pilih Buat proyek.
 - b. Dari panel navigasi kiri, pilih Proyek. Pilih Buat di sudut kanan atas untuk membuat proyek.

2. Di bagian Buat Proyek, masukkan nama Proyek, dan berikan Deskripsi opsional.
3. Pilih Buat.

example-portal

example-portal <

Home
Projects
Documentation [?](#)
Contact [?](#)

[Home](#) > [Projects](#) > Create

Create project

Create a project to host your dashboards.

Project name
Example project
The project name must contain from 1 and 256 characters. Character count: 15/256.

Project description - optional
An example project
The project description must contain from 1 and 2048 characters. Character count: 19/2048.

Cancel **Create**

Perbarui proyek

Edit proyek

1. Pilih tombol Perbarui di pojok kanan atas halaman Proyek, untuk mengedit detail proyek.
2. Ubah nama proyek dengan mengedit nama Proyek.
3. Ubah deskripsi proyek dengan mengedit detail Deskripsi.
4. Pilih Perbarui untuk menyimpan perubahan Anda.

example-portal

example-portal <

Home
Projects
Documentation [?](#)
Contact [?](#)

[Home](#) > [Projects](#) > Example-project

Example-project

[Refresh](#) [Delete](#) **[Update](#)**

Project details

Description Example project	Creation date 10/25/2024, 11:17:28 AM	Last update date 10/25/2024, 11:17:28 AM

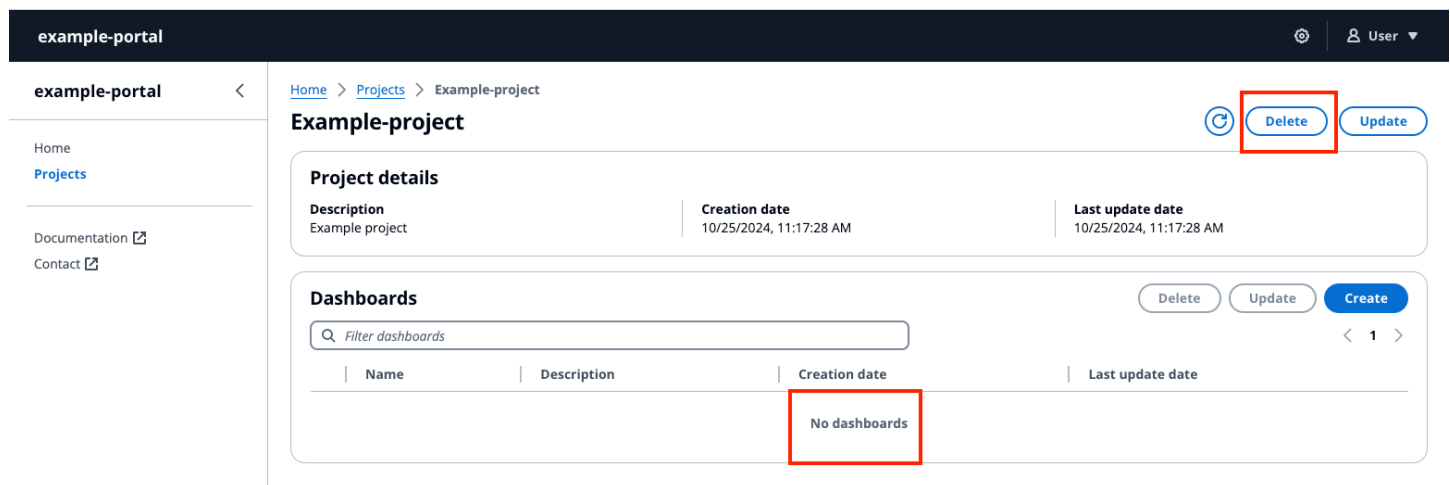
Dashboards [Delete](#) [Update](#) [Create](#)

	Name	Description	Creation date	Last update date
☒	example dashboard	dashboard	10/25/2024, 11:17:52 AM	10/25/2024, 11:17:52 AM

Hapus proyek

Hapus proyek

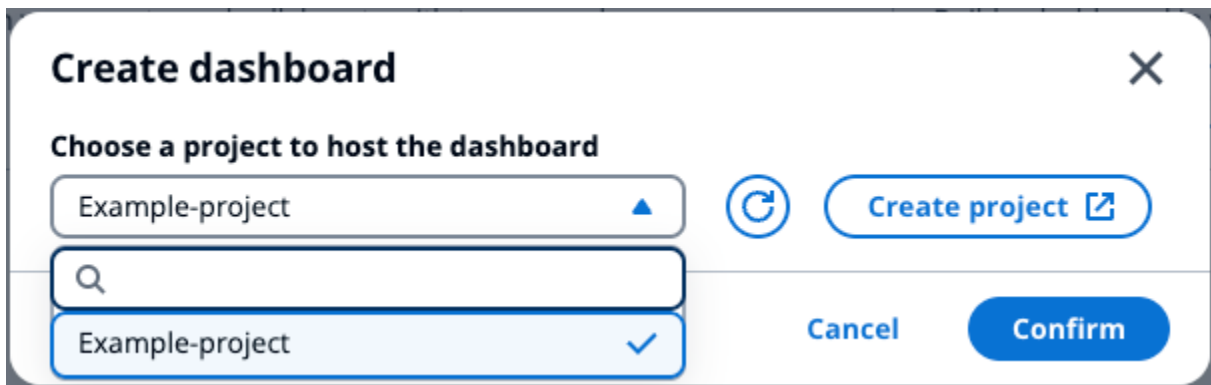
1. Anda hanya dapat menghapus proyek setelah semua dasbor dalam proyek dihapus.
2. Pilih tombol Delete di kanan atas halaman Project.
3. Konfirmasikan lagi bahwa Anda ingin menghapus proyek.
4. Pilih Hapus untuk menghapus proyek.



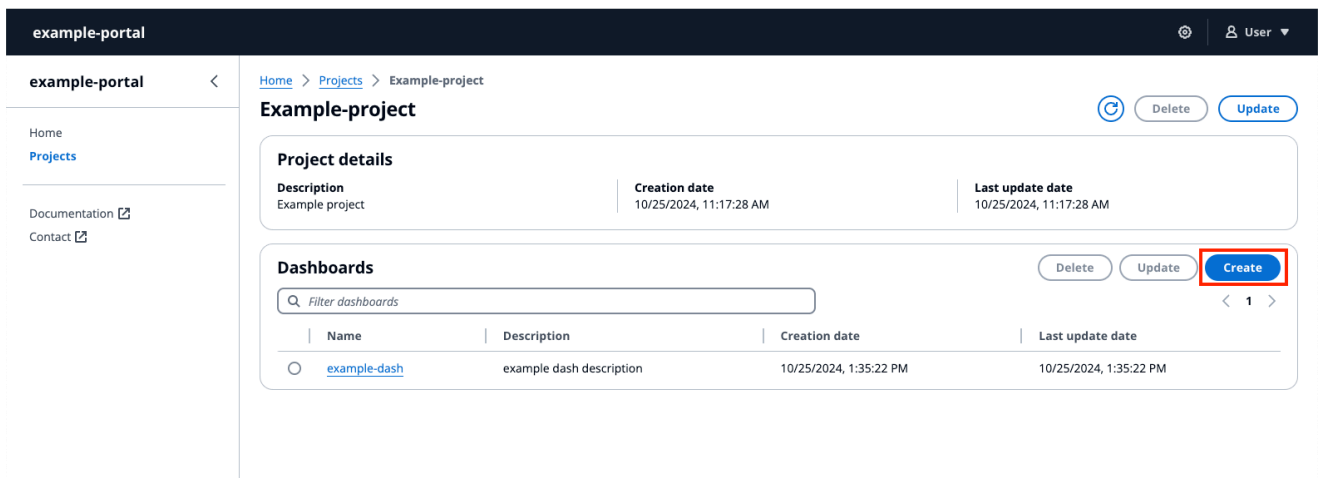
Membuat sebuah Dasbor

Membuat sebuah Dasbor

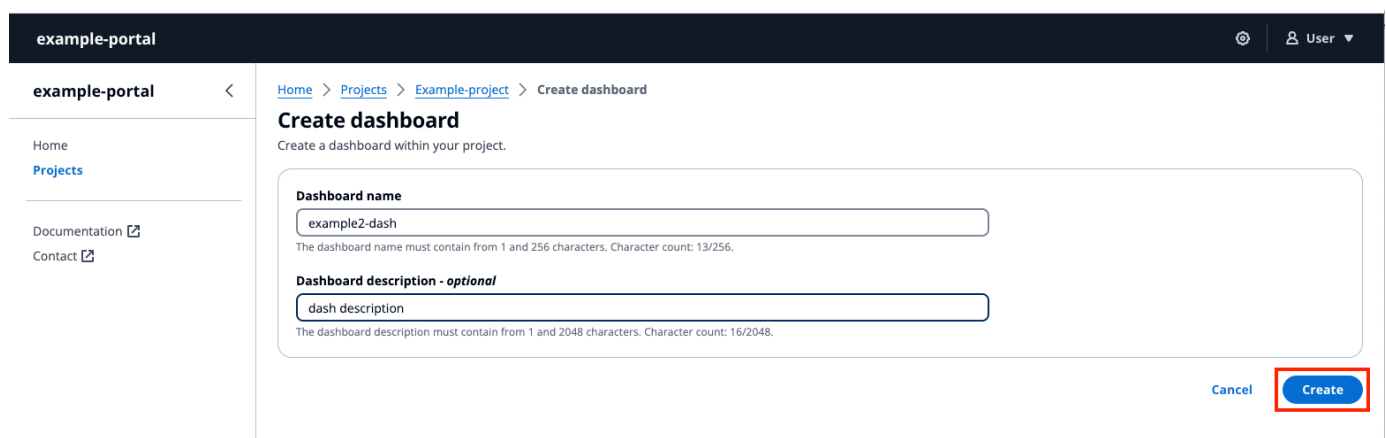
1. Buat dasbor dengan dua cara:
 - a. Buat dasbor dari Bangun dasbor di halaman Beranda.
 - i. Untuk membuat dasbor dalam proyek yang ada, pilih nama proyek dari menu tarik-turun di Pilih proyek untuk meng-host dasbor.
 - ii. Jika Anda tidak memiliki proyek, pilih Buat proyek dan pilih Konfirmasi.



- b. Buat dasbor dari proyek di bagian Proyek, di bawah Dasbor.



2. Pilih Buat di sudut kanan atas.
3. Masukkan nama Dashboard, dan berikan deskripsi Dashboard opsional.
4. Pilih Buat.



5. Konfigurasi dasbor yang baru Anda buat.

Perbarui dasbor

Bagian Dasbor mencantumkan dasbor dalam proyek. Pilih dasbor dari daftar.

Perbarui dasbor

- Pilih dasbor untuk diperbarui.

example-portal

User

Home > Projects > Example-project

Example-project

Delete Update

Project details

Description

Example project

Creation date

10/25/2024, 11:17:28 AM

Last update date

10/25/2024, 11:17:28 AM

Dashboards

Filter dashboards

Delete

Update

Create

< 1 >

Name	Description	Creation date	Last update date
example-dash	example dash description	10/25/2024, 1:35:22 PM	10/25/2024, 1:35:22 PM

- Perbarui nama Dashboard dan opsional deskripsi Dashboard. Pilih Perbarui untuk menyimpan perubahan.

Home > Projects > example-dash > Update

Update dashboard: example-dash

Dashboard name

example-update-dash-name

The dashboard name must contain from 1 and 256 characters. Character count: 24/256.

Dashboard description - optional

update dash description

The dashboard description must contain from 1 and 2048 characters. Character count: 23/2048.

Cancel

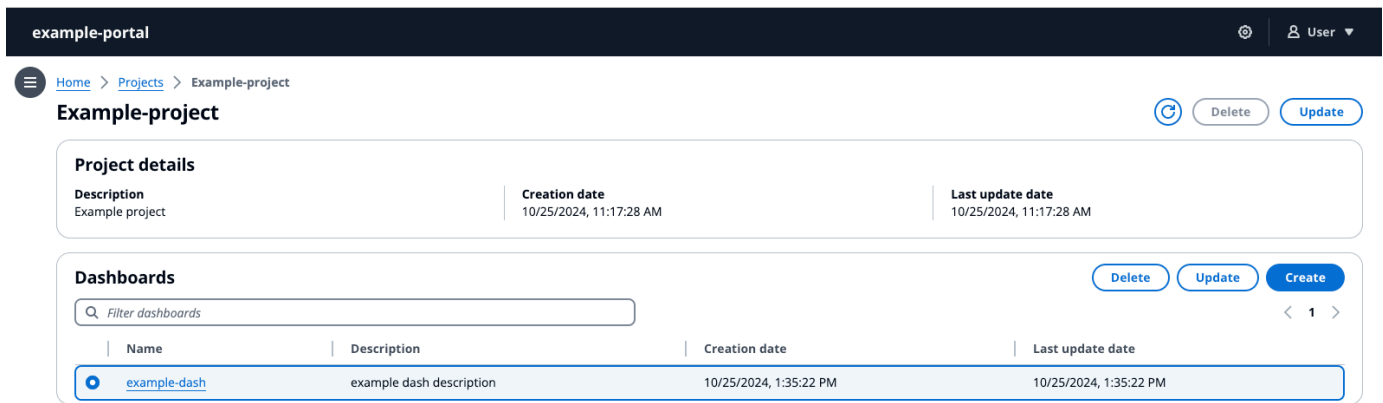
Update

Hapus dasbor

Bagian Dasbor mencantumkan dasbor dalam proyek. Pilih dasbor dari daftar.

Hapus dasbor

- Pilih dasbor untuk dihapus.



2. Pilih Hapus untuk menghapus dasbor. Ini tidak bisa dibatalkan.

Konfigurasi dasbor

Bagian Dasbor mencantumkan dasbor dalam proyek. Pilih dasbor dari daftar. Mode Edit memungkinkan Anda untuk mengkonfigurasi dasbor Anda dengan menambahkan widget dan mengonfigurasinya. Tombol Pratinjau memungkinkan Anda memvisualisasikan perubahan Anda.



Langkah-langkah untuk mengkonfigurasi dasbor Anda:

- Seret dan lepas berbagai jenis widget data ke kanvas dasbor untuk visualisasi data.
- Tambahkan data ke widget yang diinginkan, dari Resource explorer di sebelah kiri. Penjelajah Sumber Daya terdiri dari bagian aset Modeled, Unmodeled, dan Dynamic. Cari berdasarkan nama aset atau nama properti. Pilih properti yang akan ditambahkan dan pilih Tambah.
- Sempurnakan tata letak dan gaya dengan mengubah Konfigurasi pada widget. Konfigurasi komponen termasuk judul, ambang batas, dan spesifikasi konfigurasi lainnya.
- Konfigurasi rentang waktu di mana data ditampilkan.

- Pilih rentang waktu di mana data ditampilkan. Pilih rentang waktu dan kecepatan Refresh dari sudut kanan atas, dan personalisasi rentang. Pilih tingkat di mana data akan di-refresh dari menu.
- Pilih rentang waktu pada widget, dengan menggunakan roda gulir mouse trackball Anda atau Klik kanan. Ini menggerakkan rentang waktu tampilan.
- Pilih Simpan.

Topik

- [Penjelajah sumber daya](#)
- [Widget](#)
- [Konfigurasi widget](#)
- [Gunakan widget](#)
- [Alarm di widget](#)
- [AWS IoT SiteWise Asisten digunakan dalam widget](#)
- [Contoh pertanyaan untuk ditanyakan kepada AWS IoT SiteWise Asisten](#)

Penjelajah sumber daya

Bagian ini menjelaskan aset Modeled, Unmodeled, dan Dynamic. Pilih aset dari salah satu dari ketiganya dan tambahkan ke widget Anda dan visualisasikan.

Topik

- [Dimodelkan](#)
- [Tidak dimodelkan](#)
- [Aset dinamis](#)

Dimodelkan

Bagian ini menjelaskan proses pemilihan dan visualisasi aset yang dimodelkan.

Pemilihan aset

Aset dapat ditanyakan sebagai berikut:

- Cari nama aset. Gunakan wildcard*. Misalnya, Wind* mengembalikan nama aset yang dimulai dengan teksWind. Anda harus [berintegrasi dengan AWS IoT TwinMaker](#) untuk memanfaatkan fitur ini.
- Semua aset terdaftar secara default.

Dari aset yang terdaftar, filter berdasarkan nama, deskripsi, ID, atau ID model aset. Pilih satu aset untuk mencantumkan propertinya (aliran data) dan alarm.

Pemilihan aliran data

Aliran data tercantum di bawah menu Aliran Data. Filter aliran data yang tercantum oleh metadata [Properti](#) di yang terbaru//. <https://docs.aws.amazon.com/iot-sitewise/APIReference> Pilih satu atau beberapa aliran data tergantung pada widget yang dipilih.

- KPI dan Gauge hanya mendukung satu aliran data.
- Widget yang tersisa mendukung beberapa aliran data dengan multi-pilihan.

Pemilihan alarm

AWS IoT SiteWise alarm tercantum di bawah menu Aliran Data Alarm. Filter aliran data alarm yang tercantum oleh metadata alarm. nama, properti input, dan ID model komposit adalah beberapa metadata yang digunakan untuk penyaringan. Pilih satu atau beberapa aliran data tergantung pada widget yang dipilih.

- KPI dan Gauge hanya mendukung satu alarm.
- Widget yang tersisa mendukung beberapa alarm dengan multi-pilihan.

Visualisasi aset yang dimodelkan

1. Seret widget ke kanvas. Pilih properti untuk setiap panel widget untuk membangun dasbor.
2. Opsi Filter memfilter aset untuk memilih aset yang akan divisualisasikan. Pemfilteran dilakukan dengan teks, properti atau nilai. Pemfilteran adalah untuk aset yang dimuat ke browser, dan bukan pemfilteran backend.
3. Cari untuk membuat daftar aset untuk ditambahkan ke widget Anda.
4. Tambahkan aset ke widget di kanvas.
5. Pilih Reset untuk memilih aset lain, atau buat modifikasi pada aset yang dipilih.

6. Simpan dasbor. Dalam mode Pratinjau, pilih aset yang berbeda dari menu tarik-turun untuk memantau properti di bawah setiap aset tanpa merekonstruksi panel data.

 Note

Roda pengaturan konfigurasi di sisi kanan menampilkan Preferensi bagi pengguna untuk memilih seperti Ukuran halaman, kolom pertama lengket, kolom terakhir lengket, dan preferensi Kolom. Sesuaikan preferensi Anda, dan pilih Konfirmasi untuk menerapkan perubahan.

Modeled

Unmodeled

Dynamic assets

Assets (1)

Browse through your asset hierarchy and select an asset to view its associated data streams.

Root

Search

Search for resources

Search

Filter

Filter assets by text, property, or value

< 1 > ⚙

Name

Description

Demo Wind Farm Asset

Asset properties (8)

Select a modeled datastream to add to a selected widget

Filter

Filter asset properties by text, property, or value

< 1 > ⚙

	Name	Unit	Data ty...	Latest ...	Latest value time
☐	Total Aver...		DOUBLE	37478.2303	2024-10-01 09:50:00 p.m.
☐	Total Aver...		DOUBLE	6.0000	2022-10-26 03:42:43 p.m.
☐	Total Aver...		DOUBLE	555.0000	2022-10-13 11:59:49 p.m.
☐	Code		INTEGER	300.0000	2022-10-13 10:59:28 p.m.
☐	Reliability ...		STRING	Mary Major	2022-10-13 10:59:28 p.m.
☐	Location		STRING	Renton	2022-10-13 10:59:28 p.m.
☐	Total Over...		DOUBLE	900.0000	2024-10-01 09:50:00 p.m.
☐	recipient a...		STRING	54a88418-...	2022-10-26 03:42:43 p.m.

Tidak dimodelkan

Bagian ini menjelaskan pencarian aliran data yang tidak dimodelkan dan menambahkannya ke widget untuk divisualisasikan.

Konfigurasi dasbor

681

Visualisasi aliran data yang tidak dimodelkan

1. Seret widget ke kanvas. Pilih properti untuk setiap panel widget untuk membangun dasbor.
2. Aliran data yang tidak dimodelkan tercantum di bawah bagian Deret waktu. Mereka memiliki properti yang dapat disesuaikan.
3. Opsi Filter menyaring aliran data untuk memvisualisasikan. Pemfilteran adalah untuk aliran data yang dimuat ke browser, dan bukan penyaringan back end.
4. Tambahkan aliran data ke widget di kanvas.
5. Pilih Reset untuk membatalkan pilihan aliran data.
6. Simpan dasbor. Dalam mode Pratinjau, pilih aset yang berbeda dari menu tarik-turun untuk memantau properti di bawah setiap aset tanpa merekonstruksi panel data.

Note

Roda pengaturan konfigurasi di sisi kanan menampilkan Preferensi bagi pengguna untuk memilih seperti Ukuran halaman, kolom pertama lengket, kolom terakhir lengket, dan preferensi Kolom. Sesuaikan preferensi Anda, dan pilih Konfirmasi untuk menerapkan perubahan.

Modeled
Unmodeled
Dynamic assets

Time series (1)

Select a unmodeled datastream to add to a selected widget

Filter
< 1 > 

☒	Alias	ID	Data type	Latest val...	Latest valu...
☒	DemoDisass...	33e78bb9-39...	DOUBLE	3.0283	2024-10-01 0...

Aset dinamis

SiteWise Monitor baru memungkinkan pelanggan untuk secara dinamis beralih aset untuk model aset yang dipilih. Anda dapat memvisualisasikan properti dari aset yang berbeda dengan memilih dari menu tarik-turun.

Visualisasi aset dinamis

1. Pilih tab Aset dinamis pada penjelajah sumber daya.
2. Pilih model Aset untuk mencantumkan aset dari menu tarik-turun.
3. Pilih aset Default dari menu tarik-turun.
4. Pilih Setel model aset untuk memilih model aset.
5. Simpan dasbor. Dalam mode Pratinjau, pilih aset yang berbeda dari menu tarik-turun untuk memantau properti di bawah setiap aset, tanpa merekonstruksi panel data.

Resource explorer | <

Modeled | **Unmodeled** | **Dynamic assets**

Dynamic asset visualizations allow you to build one visualization to represent any asset of a specified asset model. [Learn more](#)

Asset model
Select an asset model to add the associated properties into your dynamic display.

Demo Wind Farm Asset Model 2022-Oct-14 05:59:28 ▲

🔍 *Filter asset models*

Demo Turbine Asset Model 2022-Oct-14 05:59:12
This is an asset model used in the IoT SiteWise Demo for representing a turbine in a wind farm. It will be deleted at the end of the demo.

Demo Wind Farm A Demo Turbine Asset Model 2022-Oct-14 05:59:12
This is an asset model used in the IoT SiteWise Demo for representing a wind farm. It will be deleted at the end of the demo. ✓

Finished loading asset models.

Set asset model

Widget

Widget mendukung berbagai fitur, termasuk alarm, streaming langsung berkinerja tinggi, dan sinkronisasi yang lancar dengan komponen Kit Aplikasi IoT lainnya. Dasbor mendukung widget berikut:

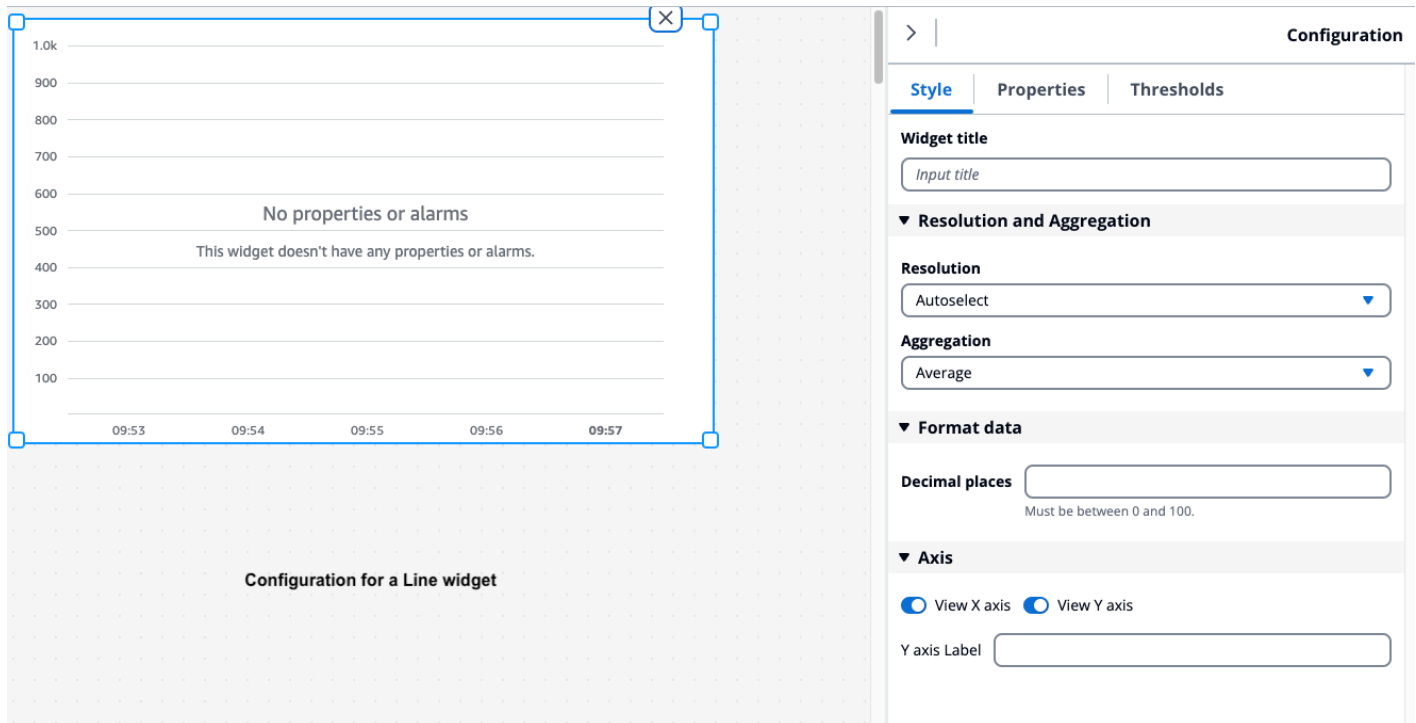
- **Line - Widget Line** adalah widget visualisasi yang menampilkan tren dan perubahan dari waktu ke waktu. Ini terdiri dari serangkaian titik data, masing-masing diwakili oleh titik atau penanda, dihubungkan oleh segmen garis lurus untuk membuat grafik garis. Ini mendukung berbagai fitur, termasuk alarm, ambang batas, streaming langsung berkinerja tinggi, dan sinkronisasi yang lancar dengan komponen Kit Aplikasi IoT lainnya. Widget ini dapat disesuaikan untuk mengkomunikasikan data yang kompleks dengan jelas dan ringkas.
- **Bar chart - Bar chart** adalah alat visualisasi yang kuat yang menampilkan data deret waktu. Ini mendukung berbagai fitur, termasuk alarm, streaming langsung berkinerja tinggi, dan sinkronisasi yang lancar dengan komponen Kit Aplikasi IoT lainnya.
- **Timeline - Widget Timeline** menyediakan cara untuk memvisualisasikan dan menavigasi data deret waktu dari sumber data. Hal ini unik untuk menampilkan nilai aliran data adalah warna yang berbeda pada timeline. Ini mendukung serangkaian fitur yang kaya termasuk alarm, streaming langsung kinerja tinggi, dan sinkronisasi yang lancar di seluruh komponen Kit Aplikasi IoT lainnya. Paling baik digunakan untuk menampilkan tipe data non-numerik/
- **KPI — Komponen Indikator Kinerja Utama (KPI)** memberikan representasi ringkas dari gambaran umum properti aset Anda. Ini mendukung alarm dan ambang batas. Ikhtisar ini memberikan wawasan penting tentang kinerja keseluruhan perangkat, peralatan, dan proses Anda. KPI hanya mendukung satu aliran data atau alarm, dan bukan beberapa aliran data.
- **Gauge - Komponen Gauge** memberikan representasi ringkas dari ikhtisar properti aset Anda. Ini digunakan untuk memvisualisasikan wawasan kritis ke dalam kinerja keseluruhan perangkat, peralatan, atau proses Anda. Secara fungsional sama dengan KPI, tetapi berbeda secara visual. Gauge menampilkan nilai aliran data, ambang batas, dan rentang nilai. Anda dapat berinteraksi dengan AWS IoT data dari satu atau lebih sumber data dengan Gauge.
- **Tabel — Komponen Tabel** menyediakan bentuk ringkas untuk melihat satu atau lebih aliran data dari satu atau lebih sumber data deret waktu. Ini menampilkan aset dengan Properti, Nilai Terbaru dan Unit dalam bentuk tabel. Mendukung AWS IoT SiteWise alarm.
- **Teks - Widget Teks** membantu menulis teks dengan berbagai warna dan font. Anda dapat membuat tautan dengan mengaitkan teks dengan URL. Bidang Properti dan Ambang tidak diaktifkan untuk widget ini.



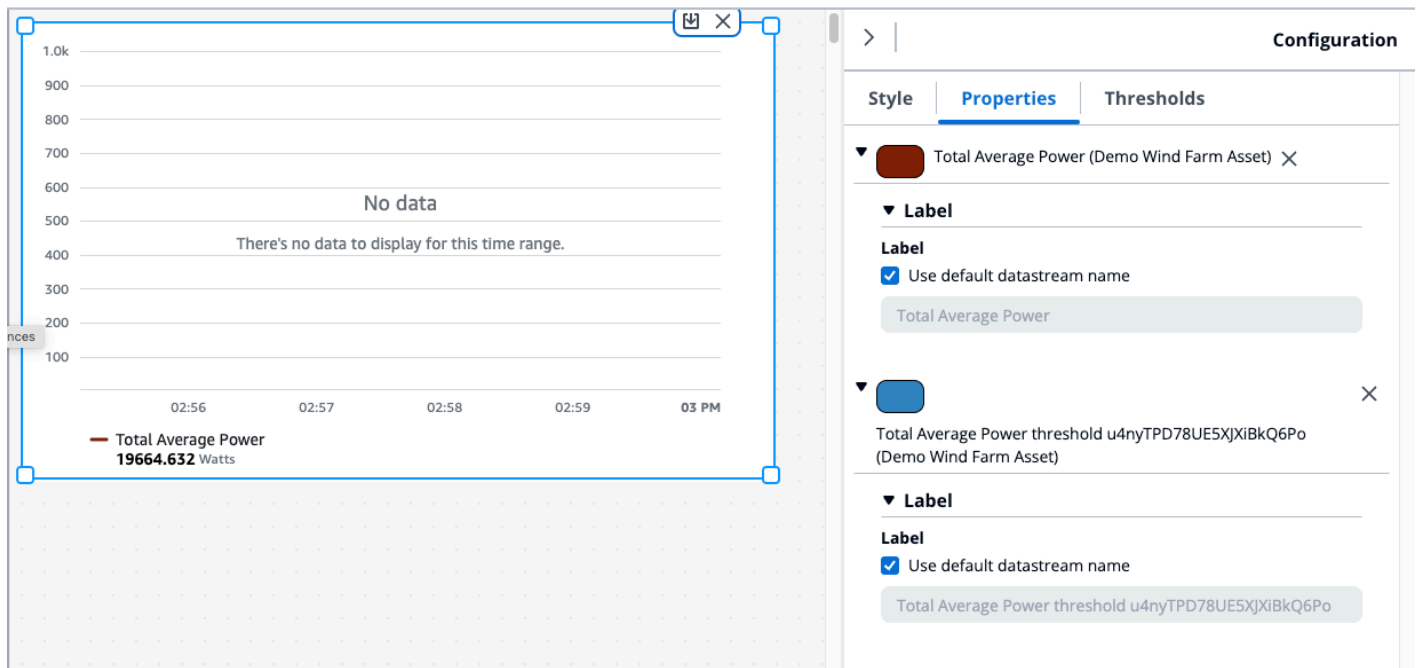
Konfigurasi widget

Setelah widget ditambahkan ke dasbor, Anda dapat mengonfigurasi widget dengan memilih ikon Konfigurasi di panel kanan.

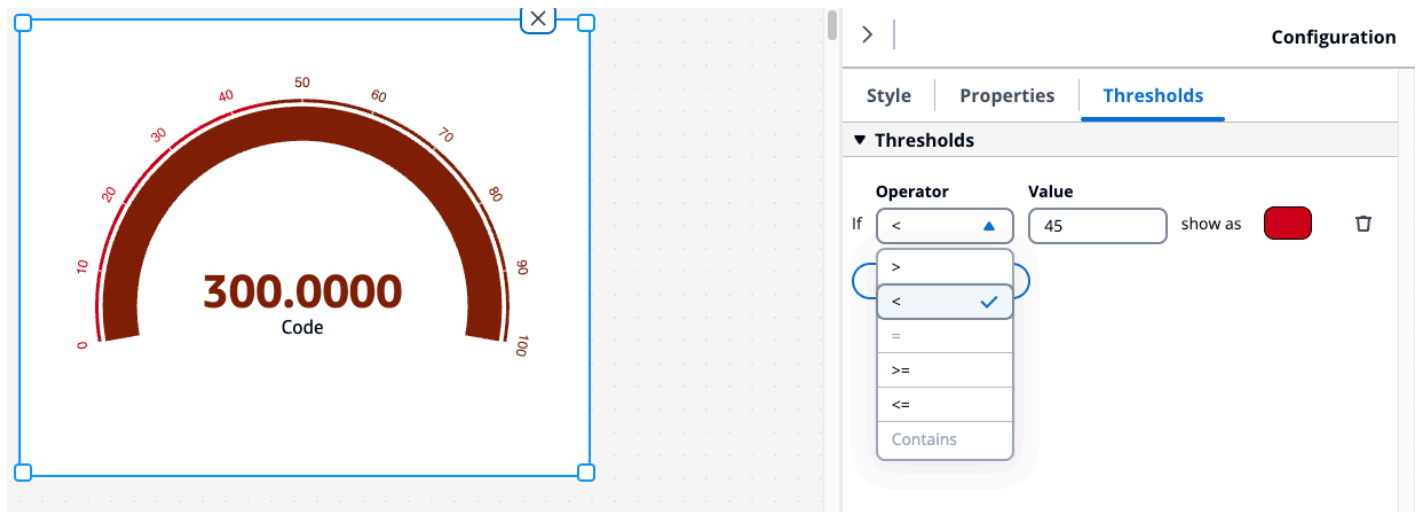
- **Style** - Tambahkan judul dalam judul Widget. Widget yang berbeda memiliki konfigurasi yang berbeda. Beberapa contoh tercantum di bawah ini.
- **Widget bar:**
 - **Resolusi dan Agregasi** - Tetapkan nilai untuk resolusi dan agregasi di sini.
 - **Format data** - Atur tempat desimal ke jumlah desimal yang akan ditampilkan.
 - **Gaya tampilan** - Pilih nilai untuk ditampilkan.
 - **Sumbu** - Pilih untuk menampilkan sumbu.
- **Widget baris:**
 - **Resolusi dan Agregasi** - Tetapkan nilai untuk resolusi dan agregasi di sini.
 - **Format data** - Atur tempat desimal ke jumlah desimal yang akan ditampilkan.
 - **Sumbu Y** - Tambahkan Label, dan nilai Min dan Maks.
 - **Gaya widget** - Pilih Jenis garis, Gaya garis, Ketebalan garis, dan nilai bentuk titik data.
 - **Legenda** — Pilih Alignment, dan Display.
- **Widget pengukur:**
 - **Resolusi dan Agregasi** - Tetapkan nilai untuk resolusi dan agregasi di sini.
 - **Format data** - Atur tempat desimal ke jumlah desimal yang akan ditampilkan.
 - **Gaya tampilan** - Pilih nilai untuk ditampilkan.
 - **Sumbu Y** - Tambahkan Label, dan nilai Min dan Maks.
 - **Font** - Pilih ukuran Font, Ukuran font Unit, dan nilai ukuran font Label.



- Properti - Semua properti widget tercantum di bagian ini. Widget yang berbeda memiliki sifat yang berbeda. Beberapa contoh tercantum di bawah ini.
- Widget baris:
 - Label - Pilih untuk menggunakan nama datastream default atau beri nama baru.
 - Gaya - Atur tipe Garis, Gaya garis ke jumlah desimal yang akan ditampilkan.
 - Sumbu Y - Pilih nilai ke gaya default, tampilkan kontrol sumbu Y dan atur nilai Min dan Maks.
- Widget tabel:
- Widget tabel:
- Label - Pilih untuk menggunakan nama datastream default atau beri nama baru.



- Ambang batas - Tambahkan Ambang untuk widget. Widget yang berbeda memiliki konfigurasi yang berbeda. Beberapa contoh tercantum di bawah ini.
- Widget bagan batang:
 - Pilih Tambahkan ambang batas untuk ditambahkan ke widget.
 - Pilih Operator dan berikan Nilai untuk ambang batas. Sesuaikan ambang batas dengan warna dari palet warna.
 - Anda dapat memilih untuk menerapkan ambang batas di semua data.
- Widget baris:
 - Pilih Tambahkan ambang batas untuk ditambahkan ke widget.
 - Pilih Operator dan berikan Nilai untuk ambang batas. Sesuaikan ambang batas dengan warna dari palet warna.
 - Pilih cara Menampilkan ambang batas dari menu tarik-turun.
- Widget pengukur:
 - Pilih Tambahkan ambang batas untuk ditambahkan ke widget.
 - Pilih Operator dan berikan Nilai untuk ambang batas. Sesuaikan ambang batas dengan warna dari palet warna.



Gunakan widget

Anda dapat menggunakan widget di dasbor satu per satu atau dengan banyak memilihnya.

Edit widget di dasbor

Pilih satu widget dan edit. Untuk mengedit beberapa widget di dasbor, Shift+Klik kiri dan pilih semua widget di dasbor. Setelah dipilih, pengguna dapat menambahkan aliran data baru, dan memodifikasi judul Widget dalam pengaturan konfigurasi Style. Judul diubah untuk semua widget di dasbor.

Klik kanan pada kanvas, dan lakukan hal berikut:

- Salin - Tambahkan salinan widget ke kanvas.
- Hapus - Hapus widget.
- Bawa ke depan - Bawa widget yang dipilih ke depan kanvas.
- Kirim ke belakang - Kirim widget yang dipilih ke bagian belakang kanvas.

Ubah ukuran widget

Ukur ulang widget satu per satu, atau dalam grup dengan memilih banyak widget di dasbor.

Untuk mengubah ukuran widget:

- Untuk mengubah ukuran widget tunggal, pilih widget, dan seret ke sudut untuk mengubah ukurannya.

- Untuk mengubah ukuran beberapa widget, pilih beberapa widget dengan Shift+Klik kiri, dan seret ke sudut untuk mengubah ukurannya.

Hapus widget di dasbor

Hapus widget satu per satu, atau dalam grup dengan memilih banyak widget di dasbor.

Untuk menghapus widget:

- Untuk menghapus satu widget, pilih widget, dan klik kanan dan pilih Hapus. Anda juga dapat memilih, dan klik X di sudut kanan atas untuk menghapus widget.
- Untuk menghapus beberapa widget, pilih beberapa widget dengan Shift+Klik kiri, lalu Klik kanan dan pilih Hapus.

Alarm di widget

Alarm memberi tahu Anda dan tim Anda saat peralatan atau proses berjalan secara sub-optimal. Kinerja optimal mesin atau proses berarti bahwa nilai untuk metrik tertentu harus berada dalam kisaran batas tinggi dan rendah. Ketika metrik ini berada di luar jangkauan operasinya, operator peralatan harus diberi tahu sehingga mereka dapat memperbaiki masalah. Alarm membantu Anda mengidentifikasi masalah dengan cepat dan memberi tahu operator untuk memaksimalkan kinerja peralatan dan proses Anda.

Anda dapat menemukan alarm yang terkait dengan aset di tab Modeled pada Resource explorer.

- Cari dan pilih aset.
- Gulir ke bawah melewati tabel Aliran Data ke bagian Aliran Data Alarm dan perluas.
- Pilih alarm di tabel Alarm dan pilih Tambah.

Topik

- [Alarm di widget yang berbeda](#)

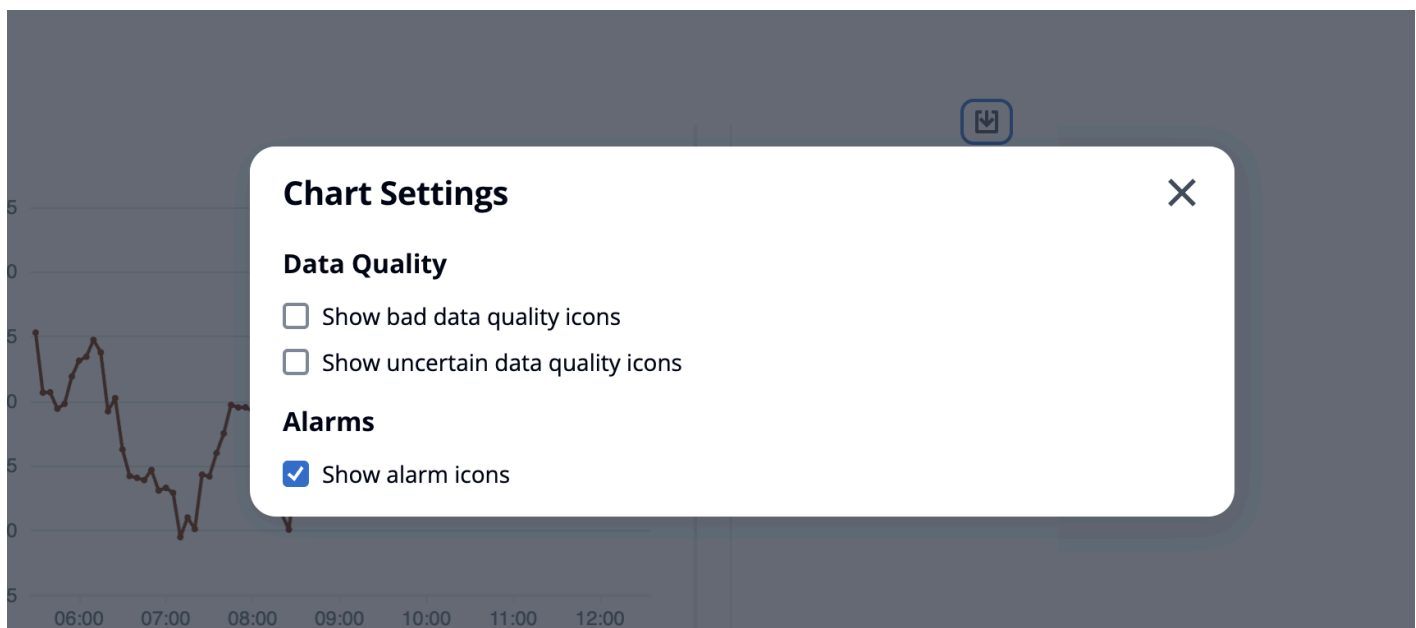
Alarm di widget yang berbeda

Untuk semua widget:

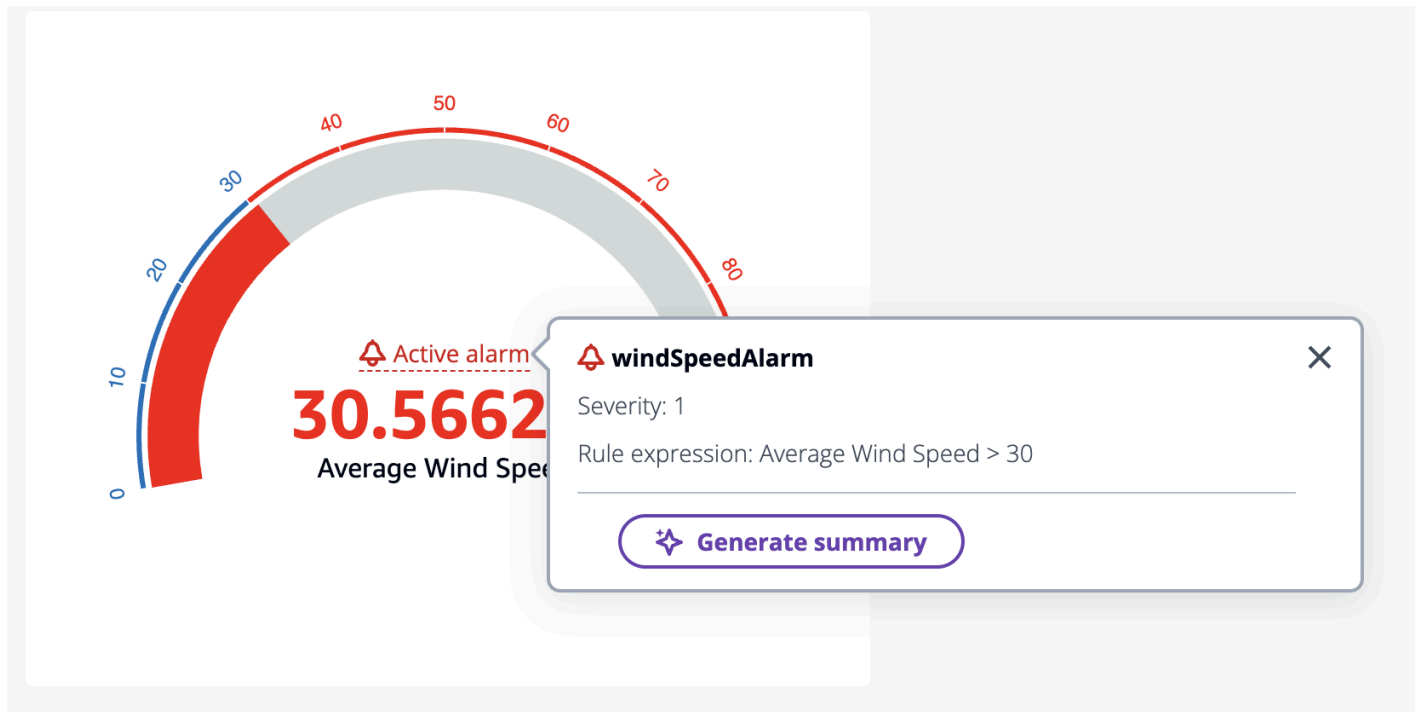
- Pengaturan properti aliran data tergantung pada jenis properti apa yang ditambahkan ke widget. Properti aliran data memiliki dukungan pengaturan properti penuh sementara properti alarm saat ini tidak mengizinkan konfigurasi pengaturan properti.
- Jika Anda menambahkan aliran data alarm, aliran data properti input terkait ditambahkan ke bagan juga. Jika Anda menghapus aliran data alarm, maka properti inputnya juga dihapus.
- Untuk mengontrol aliran data properti input alarm secara individual, Anda harus menambahkan keduanya secara terpisah.

Contoh di bawah ini menyatakan bagaimana beberapa widget menggunakan alarm.

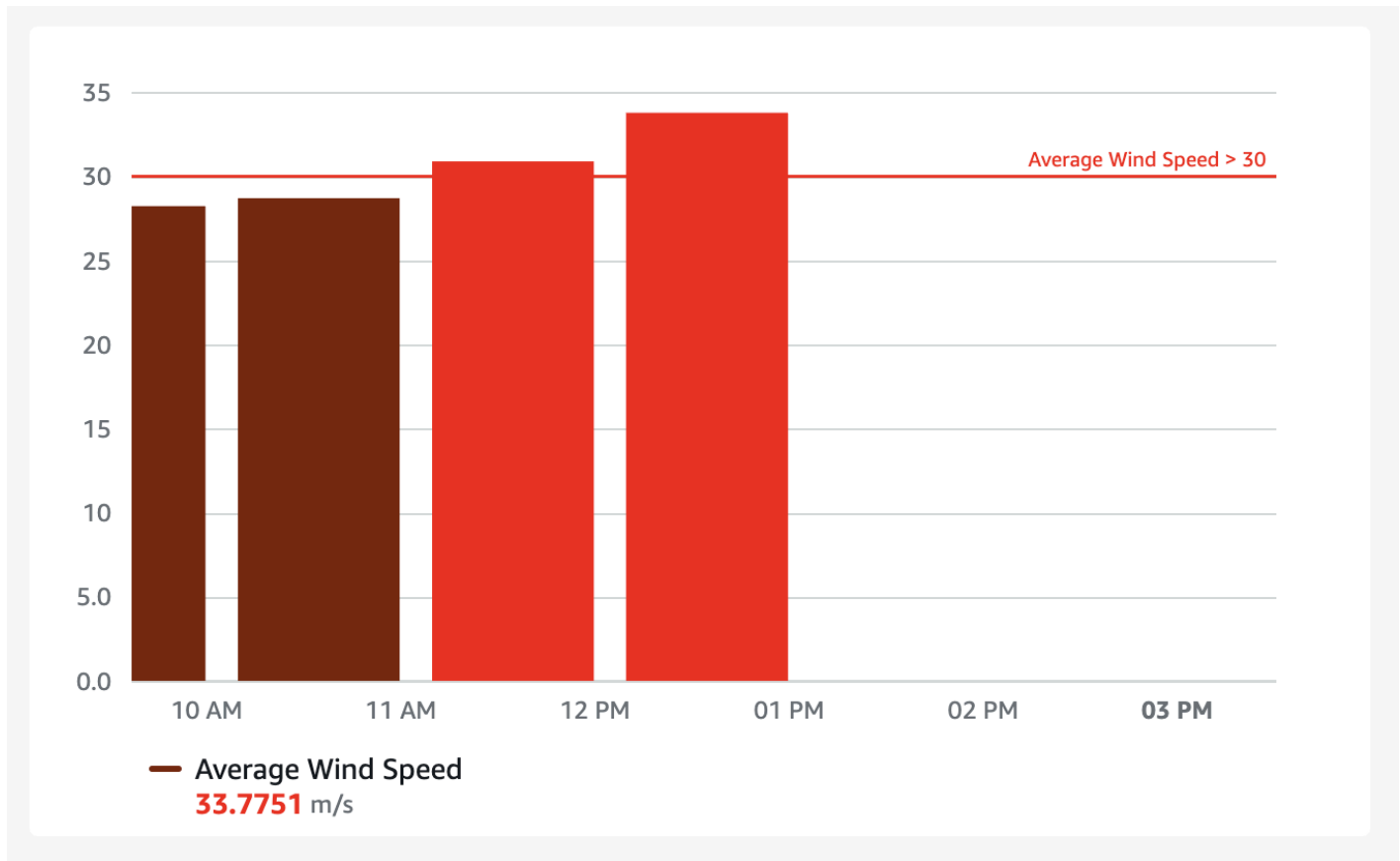
- Bagan Garis
 - Alarm dan aliran data properti inputnya ditambahkan ke bagan.
 - Anda dapat melihat status alarm di legenda bagan, dan sebagai ikon yang melayang di atas aliran data saat alarm berubah status.
 - Anda dapat menonaktifkan ikon alarm dari pengaturan bagan.



- KPI dan Gauge
 - Alarm dan aliran data properti inputnya ditambahkan ke widget yang dipilih.
 - Ambang batas alarm ditambahkan ke widget, yang berubah warna berdasarkan konfigurasinya.
 - Anda dapat memilih status alarm di widget, melihat detail alarm, dan klik Hasilkan ringkasan untuk memanggil AWS IoT SiteWise untuk mendapatkan ringkasan alarm.



- Tabel
 - Alarm dan properti inputnya ditambahkan sebagai baris di atas meja.
- Bagan batang
 - Alarm ditambahkan sebagai ambang batas ke bagan, yang mengubah warna aliran data apa pun yang melanggar ambang batas.
 - Anda dapat menambahkan aliran data terkait secara terpisah.
 - Anda tidak dapat berinteraksi dengan AWS IoT SiteWise Asisten dari widget.

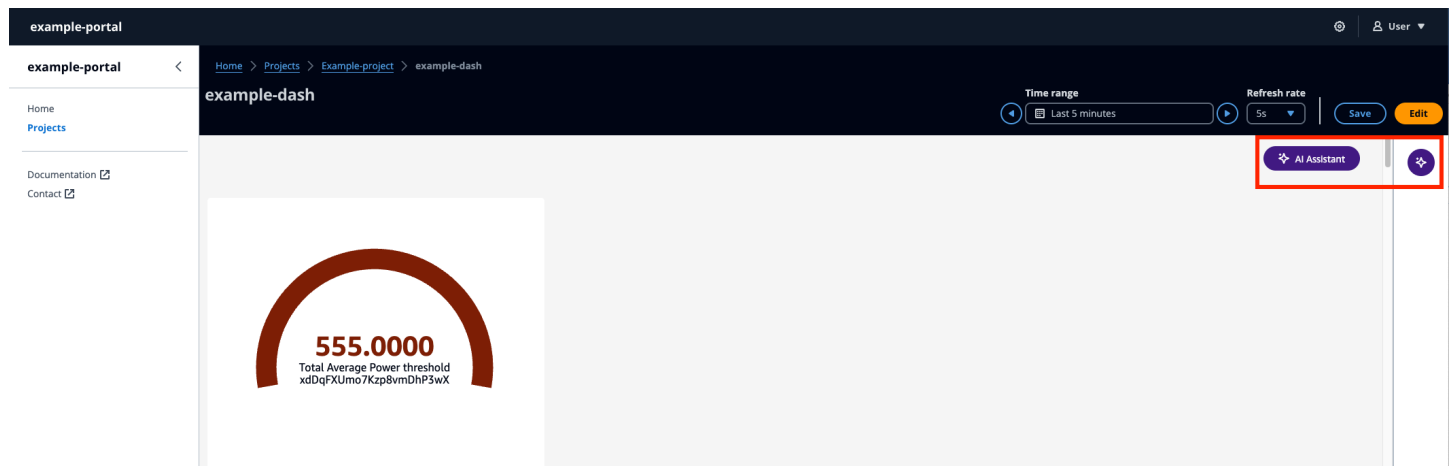


- Garis waktu status
 - Alarm ditambahkan sebagai ambang batas waktu.
 - Menambahkan status alarm dan data properti inputnya ke timeline sedang berjalan.
 - Anda tidak dapat berinteraksi dengan AWS IoT SiteWise Asisten dari widget.

AWS IoT SiteWise Asisten digunakan dalam widget

AWS IoT SiteWise Asisten adalah asisten bertenaga AI generatif. Ini memungkinkan pengguna seperti manajer pabrik, insinyur berkualitas, dan teknisi pemeliharaan untuk mendapatkan wawasan, memecahkan masalah, dan mengambil tindakan langsung dari data operasional dan perusahaan mereka. AWS IoT SiteWise Asisten mengkonsolidasikan informasi dari AWS IoT data, model aset, manual, dan dokumentasi ke dalam ringkasan peristiwa kritis yang dapat dimengerti. Ini juga memungkinkan sesi tanya jawab mendalam interaktif untuk diagnosis yang mudah, eksplorasi akar penyebab dan rekomendasi terpandu.

Tombol AWS IoT SiteWise Asisten ada di sudut kanan atas dasbor. Klik di atasnya untuk mengaktifkan Asisten. Hanya dapat digunakan dengan mode Pratinjau dasbor.



Gunakan AWS IoT SiteWise Asisten dalam skenario berikut:

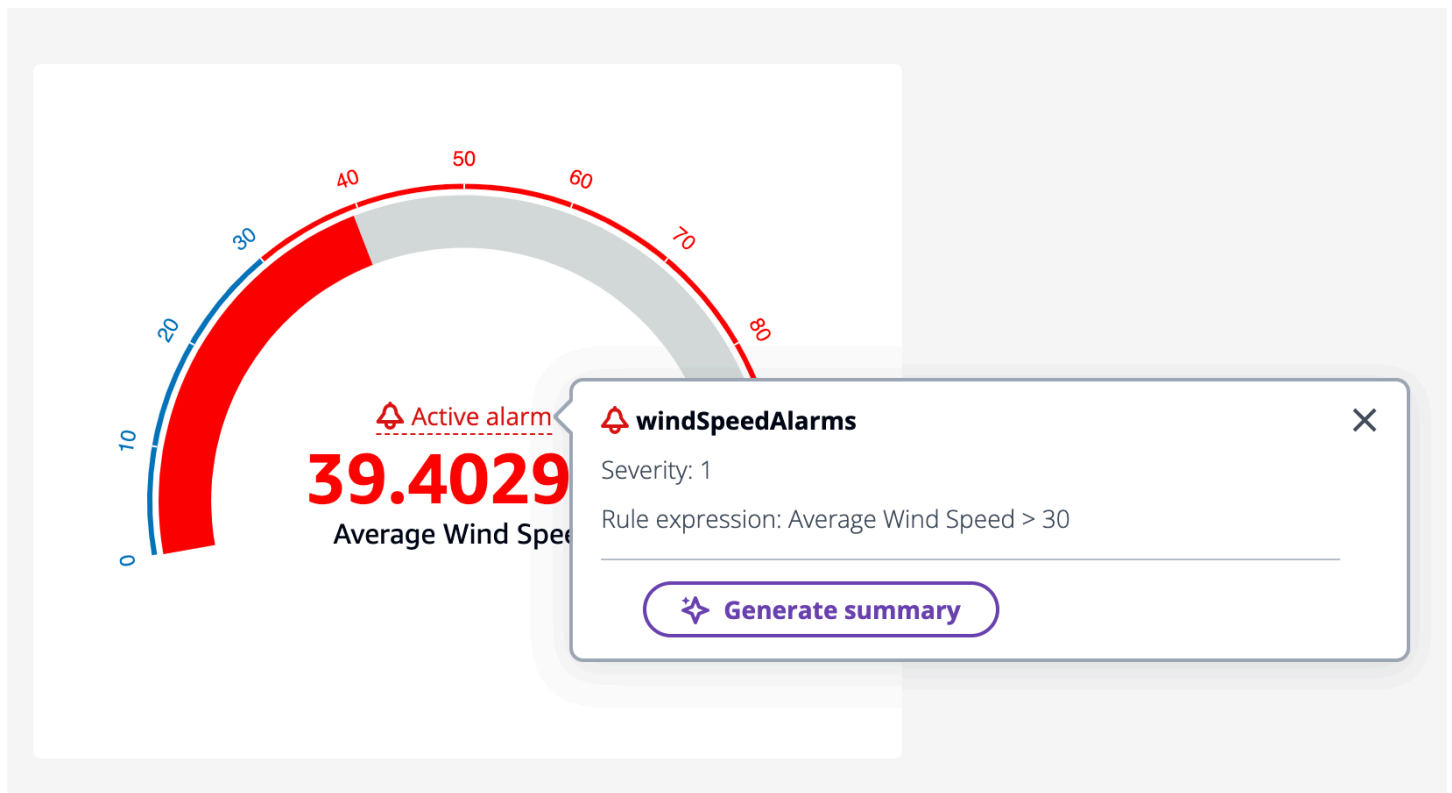
Topik

- [Kasus penggunaan - Ringkasan alarm](#)
- [Kasus penggunaan - Ringkasan situasional](#)
- [Kasus penggunaan - Ringkasan menyelam mendalam](#)

Kasus penggunaan - Ringkasan alarm

Ringkas alarm saat ini untuk panel yang dipilih di dasbor. Alarm didukung oleh widget Line, KPI, Gauge dan Table. Pilih widget dengan alarm dan rangkum.

- Pilih Alarm aktif pada widget.
- Ekspresi Keparahatan dan Aturan ditampilkan untuk alarm.
- Pilih Hasilkan ringkasan untuk menghasilkan ringkasan.



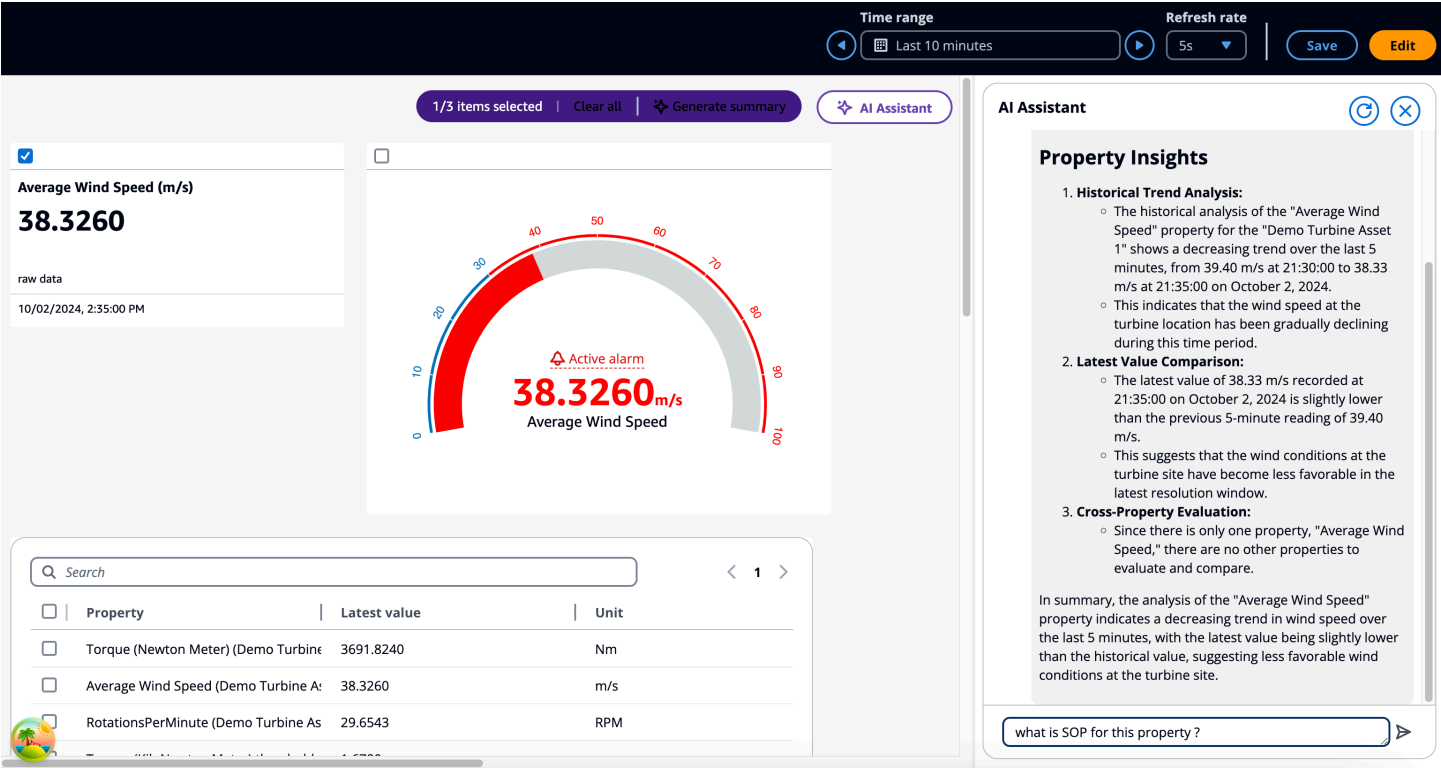
Kasus penggunaan - Ringkasan situasional

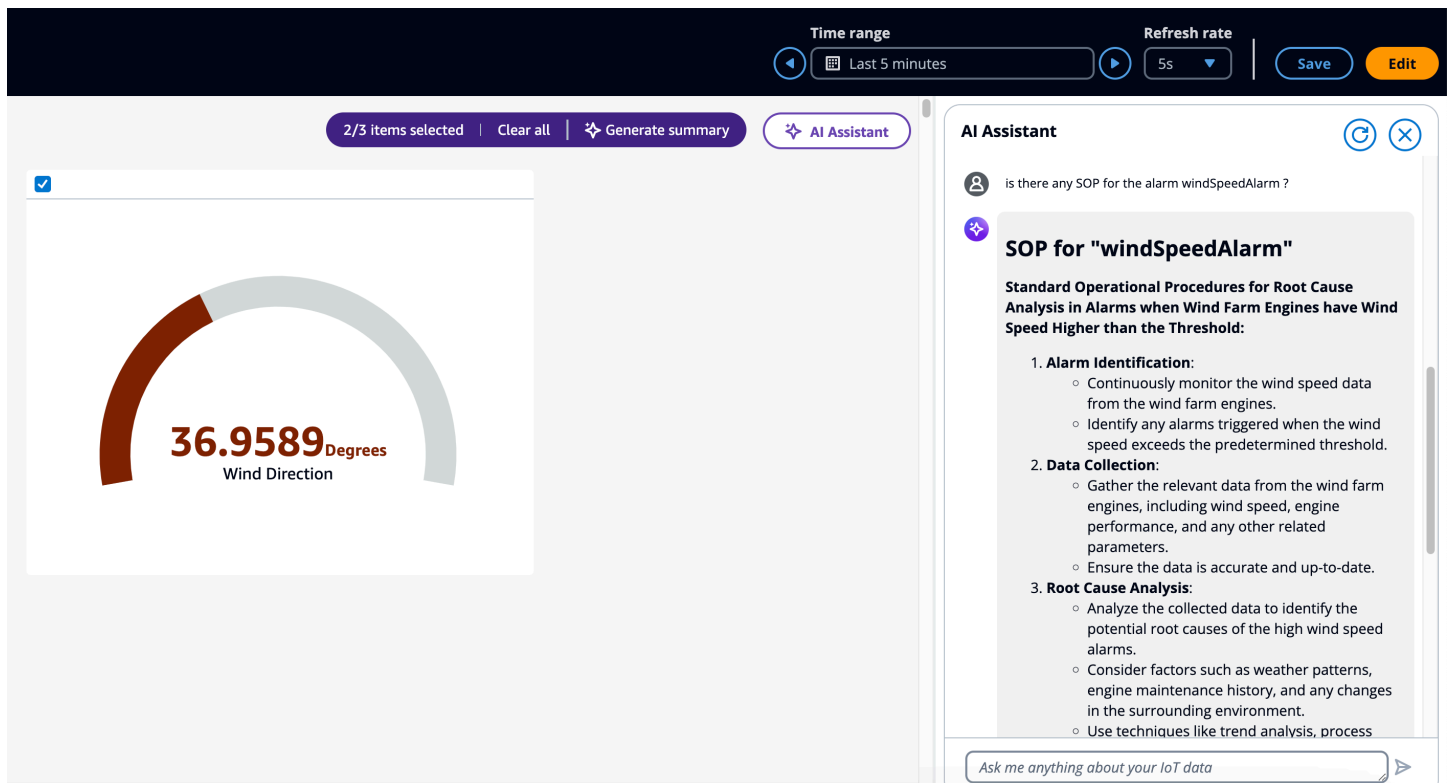
Pilih hingga tiga widget untuk diringkas. Mereka dapat menjadi kombinasi dari widget dan properti. Jika lebih dari tiga dipilih, Asisten mengembalikan kesalahan.

Buat ringkasan situasi dengan AWS IoT SiteWise Asisten

1. Klik pada Asisten AI. Ini menampilkan menu dengan tiga opsi.
 - a. Item yang dipilih - Pilih hanya tiga. Anda tidak dapat memilih lebih dari tiga.
 - b. Hapus semua — Hapus pilihan Anda.
 - c. Hasilkan ringkasan - Buat ringkasan tentang item yang dipilih.
2. Pilih Hasilkan ringkasan untuk menghasilkan ringkasan tentang item yang dipilih.

Gambar di bawah ini memiliki widget yang dipilih dan ringkasan dari AWS IoT SiteWise Asisten.





Contoh pertanyaan untuk ditanyakan kepada AWS IoT SiteWise Asisten

Note

- AWS IoT SiteWise Asisten harus menggunakan kumpulan data dengan indeks [Amazon Kendra](#) untuk pengetahuan dan panduan tingkat perusahaan. Jika Anda tidak memiliki indeks Amazon Kendra, lihat [Membuat indeks](#) untuk membuatnya. Menambahkan [kumpulan data](#) meningkatkan kualitas respons Asisten. Lihat [Membuat kumpulan data](#) untuk mempelajari lebih lanjut.
- Beberapa pertanyaan membutuhkan AWS IoT TwinMaker integrasi. Lihat [Mengintegrasikan AWS IoT TwinMaker dan AWS IoT SiteWise](#) untuk detailnya.

Beberapa pertanyaan tindak lanjut untuk ditanyakan kepada Asisten setelah mendapatkan ringkasan alarm di dasbor, sebagai bagian dari percakapan yang sama.

- Tampilkan detail aset dari ringkasan di atas?
- Apa jalur hierarkis dari root ke aset yang disebutkan?
- Apa aset keturunan dependen dari aset yang disebutkan?

- Apa aset dependen dari aset yang disebutkan yang memiliki alarm aktif?
- Temukan semua aset yang memiliki alarm aktif.

Beberapa pertanyaan tindak lanjut untuk ditanyakan kepada Asisten setelah mendapatkan ringkasan properti di dasbor, sebagai bagian dari percakapan yang sama.

- Lakukan analisis yang sama selama 24 jam terakhir.
- Temukan dokumentasi yang terkait dengan properti yang disebutkan di atas.
- Berikan rincian id aset 1da67d28-14f8-4f71-a06a-386f0425a21d/nama aset Demo Turbine Asset 1.

Panggil AWS IoT SiteWise Asisten dari API.

- Hasilkan ringkasan alarm untuk nama alarm windSpeedAlarm di id aset d591e153-e5cf-4206-96bb-ce3c119d9d2d.
- Hasilkan ringkasan alarm selama 12 jam terakhir/2 hari/1 minggu untuk nama windSpeedAlarm di id aset. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Hasilkan ringkasan properti untuk id properti ab187fb7-d74b-44d9-bd9b-f2f19a9137cc di asset id d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Buat ringkasan properti selama 12 jam terakhir/2 hari/1 minggu untuk id properti di id ab187fb7-d74b-44d9-bd9b-f2f19a9137cc aset. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Temukan aset dengan nama aset Turbin.
- Beri saya nilai properti saat ini dari id properti 5356168c-3390-456f-802c-9f6e047810d4 di id aset d591e153-e5cf-4206-96bb-ce3c119d9d2d, 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Apa hubungan antara id aset d591e153-e5cf-4206-96bb-ce3c119d9d2d dan id aset 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Temukan dokumentasi tentang cara memperbaiki masalah RPM rendah turbin angin.
- Hasilkan ringkasan properti untuk alias WindSpeedproperti.
- Apa pemeriksaan pra-operasi menurut basis pengetahuan saya?

Data kueri dari AWS IoT SiteWise

Anda dapat menggunakan operasi AWS IoT SiteWise API untuk menanyakan nilai saat ini, nilai historis, dan agregat properti aset Anda selama interval waktu tertentu.

Gunakan fitur ini untuk mendapatkan wawasan tentang data Anda. Misalnya, temukan semua aset Anda dengan nilai properti tertentu atau buat representasi kustom data Anda. Anda juga dapat menggunakan operasi API untuk mengembangkan solusi perangkat lunak yang terintegrasi dengan data industri yang disimpan dalam AWS IoT SiteWise aset Anda. Anda juga dapat menjelajahi data aset Anda secara langsung AWS IoT SiteWise Monitor. Untuk mempelajari cara mengkonfigurasi SiteWise Monitor, lihat [Memantau data dengan AWS IoT SiteWise Monitor](#).

Operasi yang dijelaskan dalam bagian ini mengembalikan objek nilai properti yang berisi struktur stempel waktu, kualitas, nilai (TQV):

- `timestamp` Berisi waktu epoch Unix saat ini dalam hitungan detik dengan offset nanodetik.
- `quality` Berisi salah satu string berikut yang menunjukkan kualitas titik data:
 - `GOOD`— Data tidak terpengaruh oleh masalah apa pun.
 - `BAD`— Data dipengaruhi oleh masalah seperti kegagalan sensor.
 - `UNCERTAIN`— Data dipengaruhi oleh masalah seperti ketidakakuratan sensor.
- `value` Berisi salah satu bidang berikut, tergantung pada jenis properti:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

Topik

- [Kueri nilai properti aset saat ini di AWS IoT SiteWise](#)
- [Kueri nilai properti aset historis di AWS IoT SiteWise](#)
- [Agregat properti aset kueri di AWS IoT SiteWise](#)
- [AWS IoT SiteWise bahasa query](#)

Kueri nilai properti aset saat ini di AWS IoT SiteWise

Tutorial ini menunjukkan dua cara untuk mendapatkan nilai saat ini dari properti aset. Anda dapat menggunakan AWS IoT SiteWise konsol atau menggunakan API di AWS Command Line Interface (AWS CLI).

Topik

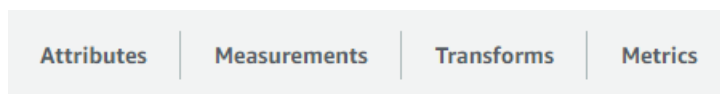
- [Kueri nilai saat ini properti aset \(konsol\)](#)
- [Kueri nilai saat ini properti aset \(AWS CLI\)](#)

Kueri nilai saat ini properti aset (konsol)

Anda dapat menggunakan AWS IoT SiteWise konsol untuk melihat nilai saat ini dari properti aset.

Untuk mendapatkan nilai saat ini dari properti aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset dengan properti untuk kueri.
4. Pilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.
5. Pilih tab untuk jenis properti. Misalnya, pilih Pengukuran untuk melihat nilai saat ini dari properti pengukuran.



6. Temukan properti untuk dilihat. Nilai saat ini muncul di kolom Nilai terbaru.

Kueri nilai saat ini properti aset (AWS CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk menanyakan nilai saat ini dari properti aset.

Gunakan [GetAssetPropertyValue](#) operasi untuk menanyakan nilai saat ini properti aset.

Untuk mengidentifikasi properti aset, tentukan salah satu dari berikut ini:

- Properti `propertyId` aset yang dikirimkan data ke. `assetId`

- ThepropertyAlias, yang merupakan alias aliran data (misalnya,/company/windfarm/3/turbine/7/temperature). Untuk menggunakan opsi ini, Anda harus terlebih dahulu menetapkan alias properti aset Anda. Untuk mengatur alias properti, lihat[Mengelola aliran data untuk AWS IoT SiteWise](#).

Untuk mendapatkan nilai saat ini dari properti aset (AWS CLI)

- Jalankan perintah berikut untuk mendapatkan nilai properti aset saat ini. Ganti *asset-id* dengan ID aset dan *property-id* dengan ID properti.

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id property-id
```

Operasi mengembalikan respons yang berisi TQV saat ini dari properti dalam format berikut.

```
{  
  "propertyValue": {  
    "value": {  
      "booleanValue": Boolean,  
      "doubleValue": Number,  
      "integerValue": Number,  
      "stringValue": "String",  
      "nullValue": {  
        "valueType": "String"  
      }  
    },  
    "timestamp": {  
      "timeInSeconds": Number,  
      "offsetInNanos": Number  
    },  
    "quality": "String"  
  }  
}
```

Kueri nilai properti aset historis di AWS IoT SiteWise

Anda dapat menggunakan [GetAssetPropertyValueHistory](#) operasi AWS IoT SiteWise API untuk menanyakan nilai historis properti aset.

Untuk mengidentifikasi properti aset, tentukan salah satu dari berikut ini:

- Properti `propertyId` aset yang dikirimkan data ke. `assetId`
- `ThepropertyAlias`, yang merupakan alias aliran data (misalnya, `/company/windfarm/3/turbine/7/temperature`). Untuk menggunakan opsi ini, Anda harus terlebih dahulu menetapkan alias properti aset Anda. Untuk mengatur alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Lulus parameter berikut untuk menyempurnakan hasil Anda:

- `startDate`—Awal eksklusif dari rentang dari mana untuk menanyakan data historis, dinyatakan dalam detik dalam waktu zaman Unix.
- `endDate`—Akhir inklusif dari rentang dari mana untuk menanyakan data historis, dinyatakan dalam detik dalam waktu zaman Unix.
- `maxResults`—Jumlah maksimum hasil untuk dikembalikan dalam satu permintaan. Default untuk hasil. 20
- `nextToken`—Token pagination dikembalikan dari panggilan sebelumnya dari operasi ini.
- `timeOrdering`—Urutan untuk diterapkan pada nilai yang dikembalikan: `ASCENDING` atau `DESCENDING`.
- `qualities`—Kualitas untuk memfilter hasil dengan: `GOOD`, `BAD`, atau `UNCERTAIN`.

Untuk menanyakan riwayat nilai untuk properti aset (AWS CLI)

1. Jalankan perintah berikut untuk mendapatkan riwayat nilai untuk properti aset. Perintah ini menanyakan riwayat properti selama interval 10 menit tertentu. Ganti *asset-id* dengan ID aset dan *property-id* dengan ID properti. Ganti parameter tanggal dengan interval untuk kueri.

```
aws iotsitewise get-asset-property-value-history \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575216600
```

Operasi mengembalikan respon yang berisi TQVs historis properti dalam format berikut:

```
{
  "assetPropertyValueHistory": [
```

```
{
  "value": {
    "booleanValue": Boolean,
    "doubleValue": Number,
    "integerValue": Number,
    "stringValue": "String",
    "nullValue": {
      "valueType": "String"
    }
  },
  "timestamp": {
    "timeInSeconds": Number,
    "offsetInNanos": Number
  },
  "quality": "String"
},
"nextToken": "String"
}
```

2. Jika ada lebih banyak entri nilai, Anda dapat meneruskan token pagination dari `nextToken` bidang ke panggilan berikutnya ke operasi. [GetAssetPropertyValueHistory](#)

Agregat properti aset kueri di AWS IoT SiteWise

AWS IoT SiteWise secara otomatis menghitung nilai properti aset agregat, yang merupakan sekumpulan metrik dasar yang dihitung selama beberapa interval waktu. AWS IoT SiteWise menghitung agregat berikut setiap menit, jam, dan hari untuk properti aset Anda:

- rata-rata — Rata-rata (mean) dari nilai properti selama interval waktu.
- count — Jumlah titik data untuk properti selama interval waktu.
- maksimum — Maksimum nilai properti selama interval waktu.
- minimum — Minimal nilai properti selama interval waktu.
- standar deviasi — Standar deviasi nilai properti selama interval waktu.
- jumlah — Jumlah nilai properti selama interval waktu.

Untuk properti non-numerik, seperti string dan Boolean, hanya menghitung agregat AWS IoT SiteWise hitungan.

Anda juga dapat menghitung metrik kustom untuk data aset Anda. Dengan properti metrik, Anda menentukan agregasi yang spesifik untuk operasi Anda. Properti metrik menawarkan fungsi agregasi tambahan dan interval waktu yang tidak dihitung sebelumnya untuk API. AWS IoT SiteWise Untuk informasi selengkapnya, lihat [Data agregat dari properti dan aset lainnya \(metrik\)](#).

Topik

- [Agregat untuk properti aset \(API\)](#)
- [Agregat untuk properti aset \(\)AWS CLI](#)

Agregat untuk properti aset (API)

Gunakan AWS IoT SiteWise API untuk mendapatkan agregat untuk properti aset.

Gunakan [GetAssetPropertyAggregates](#) operasi untuk menanyakan agregat properti aset.

Untuk mengidentifikasi properti aset, tentukan salah satu dari berikut ini:

- Properti `propertyId` aset yang dikirimkan data ke. `assetId`
- `ThepropertyAlias`, yang merupakan alias aliran data (misalnya, `/company/windfarm/3/turbine/7/temperature`). Untuk menggunakan opsi ini, Anda harus terlebih dahulu menetapkan alias properti aset Anda. Untuk mengatur alias properti, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Anda harus melewati parameter yang diperlukan berikut:

- `aggregateTypes`— Daftar agregat untuk diambil. Anda dapat menentukan salah satu dari `AVERAGE`, `COUNT`, `MAXIMUM`, `MINIMUM`, `STANDARD_DEVIATION`, dan `SUM`.
- `resolution`— Interval waktu untuk mengambil metrik: `1m` (1 menit), `15m` (15 menit), `1h` (1 jam), atau `1d` (1 hari).
- `startDate`— Awal eksklusif dari rentang dari mana untuk menanyakan data historis, dinyatakan dalam detik dalam waktu zaman Unix.
- `endDate`— Akhir inklusif dari rentang dari mana untuk menanyakan data historis, dinyatakan dalam detik dalam waktu zaman Unix.

Anda juga dapat meneruskan salah satu parameter berikut untuk menyempurnakan hasil Anda:

- **maxResults**— Jumlah maksimum hasil untuk dikembalikan dalam satu permintaan. Default untuk hasil. 20
- **nextToken**— Token pagination dikembalikan dari panggilan sebelumnya dari operasi ini.
- **timeOrdering**— Urutan untuk diterapkan pada nilai yang dikembalikan: ASCENDING atau DESCENDING.
- **qualities**— Kualitas untuk memfilter hasil dengan: GOOD, BAD, atau UNCERTAIN.

Note

[GetAssetPropertyAggregates](#) Operasi mengembalikan TQV dengan format yang berbeda dari operasi lain yang dijelaskan dalam bagian ini. **valueStruktur** berisi bidang untuk masing-masing permintaan. **aggregateTypes** **timestamp** Berisi waktu agregasi terjadi, dalam detik dalam waktu epoch Unix.

Agregat untuk properti aset ()AWS CLI

Untuk menanyakan agregat untuk properti aset ()AWS CLI

1. Jalankan perintah berikut untuk mendapatkan agregat untuk properti aset. Perintah ini menanyakan rata-rata dan jumlah dengan resolusi 1 jam untuk interval 1 jam tertentu. Ganti **asset-id** dengan ID aset dan **property-id** dengan ID properti. Ganti parameter dengan agregat dan interval untuk kueri.

```
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
```

Operasi mengembalikan respon yang berisi TQVs historis properti dalam format berikut. Respons hanya mencakup agregat yang diminta.

```
{
  "aggregatedValues": [
```



```
{
  "timestamp": Number,
  "quality": "String",
  "value": {
    "average": Number,
    "count": Number,
    "maximum": Number,
    "minimum": Number,
    "standardDeviation": Number,
    "sum": Number
  }
},
"nextToken": "String"
}
```

2. Jika ada lebih banyak entri nilai, Anda dapat meneruskan token pagination dari `nextToken` bidang ke panggilan berikutnya ke operasi. [GetAssetPropertyAggregates](#)

Note

Jika rentang kueri berisi `null` nilai TQVs, lihat [AssetPropertyValueAPI](#). Semua statistik kecuali hitungan, menghasilkan `null` respons, mirip dengan statistik untuk `String` TQVs. Jika rentang kueri Anda berisi `Double.NaN` tipe ganda TQVs, semua perhitungan kecuali hitungan akan menghasilkan `aDouble.NaN`.

AWS IoT SiteWise bahasa query

Dengan operasi [ExecuteQueryAPI](#) pengambilan AWS IoT SiteWise data, Anda dapat mengambil informasi tentang definisi struktural deklaratif, dan data timeseries yang terkait dengannya, dari berikut ini:

- model
- aset
- ukuran
- metrik
- berubah

- agregat

Hal ini dapat dilakukan dengan SQL seperti pernyataan query, dalam satu permintaan API.

Note

Fitur ini tersedia di semua Wilayah di mana keduanya AWS IoT SiteWise dan AWS IoT TwinMaker tersedia, kecuali AWS GovCloud (AS-Barat).

Topik

- [Prasyarat](#)
- [Referensi bahasa kueri untuk AWS IoT SiteWise](#)

Prasyarat

AWS IoT SiteWise memerlukan izin untuk berintegrasi AWS IoT TwinMaker sehingga dapat mengatur dan memodelkan data industri.

Sebelum Anda dapat mengambil informasi tentang model, aset, pengukuran, metrik, transformasi, dan agregat, pastikan prasyarat berikut terpenuhi:

- Peran terkait layanan untuk keduanya AWS IoT SiteWise dan AWS IoT TwinMaker disiapkan di akun Anda AWS . Untuk informasi selengkapnya tentang peran terkait layanan, lihat [Membuat peran terkait layanan](#) di Panduan Pengguna IAM.
- AWS IoT SiteWise Integrasi yang diaktifkan untuk peran IAM Anda. Untuk informasi selengkapnya, lihat [Integrasikan AWS IoT SiteWise dan AWS IoT TwinMaker](#).
- AWS IoT TwinMaker Ruang kerja dengan ID `IoTSiteWiseDefaultWorkspace` di akun Anda di Wilayah. Untuk informasi selengkapnya, lihat [Menggunakan](#) Panduan AWS IoT TwinMaker Pengguna. `IoTSiteWiseDefaultWorkspace`
- Baik mode harga bundel standar atau berjenjang untuk AWS IoT TwinMaker diaktifkan. Untuk informasi selengkapnya, lihat [AWS IoT TwinMaker Mengganti mode harga](#) di Panduan AWS IoT TwinMaker Pengguna.

Referensi bahasa kueri untuk AWS IoT SiteWise

AWS IoT SiteWise mendukung bahasa kueri yang kaya untuk bekerja dengan data Anda. Tipe data, operator, fungsi, dan konstruksi yang tersedia dijelaskan dalam topik berikut.

Lihat [Kueri contoh](#) untuk menulis kueri dengan bahasa AWS IoT SiteWise kueri.

Topik

- [Memahami tampilan referensi kueri](#)
- [Jenis data yang didukung](#)
- [Mengambil data dengan pernyataan SELECT](#)
- [Operator logis](#)
- [Operator perbandingan](#)
- [Kueri contoh](#)

Memahami tampilan referensi kueri

Bagian ini memberikan informasi untuk membantu Anda memahami tampilan AWS IoT SiteWise, seperti metadata proses dan data telemetri.

Tabel berikut memberikan nama tampilan dan deskripsi tampilan.

Model data

Lihat nama	Lihat deskripsi
aset	Berisi informasi tentang aset dan model derivasi.
aset_property	Berisi informasi tentang struktur properti aset.
raw_time_seri	Berisi data historis deret waktu.
terbaru_value_time_series	Berisi nilai terbaru dari deret waktu.
precomputed_aggregates	Berisi nilai properti aset agregat yang dihitung secara otomatis. Mereka adalah satu set metrik dasar yang dihitung selama beberapa interval waktu.

Tampilan berikut mencantumkan nama kolom untuk kueri bersama dengan data sampel.

Lihat:aset

aset_id	asset_name	asset_description	asset_model_id
88898498-0b8b-42b5-bf57-16180bc3d3a0	WindTurbine A	WindTurbine Aset A	17847250-5bf0-4f74-b775-cc03f05e7cb8
17847250-5bf0-4f74-b775-cc03f05e7cb8	Model Aset Turbin Angin	Merupakan turbin di ladang angin.	

Lihat:ASET_PROPERTY

property_id	aset_id	property_name	property_alias	asset_composite_model_id	
b29be434-b000-4d74-b809-75287d83bcd6	88898498-0b8b-42b5-bf57-16180bc3d3a0	suhu motor	Rochester2/44///Line-5/Bus-2/Machine-5/Temperature		
3b458f00-24e7-458a-b4e8-c6026eff654a	88898498-0b8b-42b5-bf57-16180bc3d3a0	arah angin	/company/windfarm/3/turbine/7/winddirection	2f458n00-56e7-458h-b4e8-c6026eff985g	

Lihat:RAW_TIME_SERIES

aset_id	property_id	property_alias	event_timestamp	kualitas	boolean_nilai	int_nilai	nilai_ganda	string_nilai
88898498-0b8b-42b5-bf57-16180bc3d3a0	b29be434-b000-4d74-b809-75287d83bcd6	Rochester2/44///	157521960	BAGUS			115.0	

aset_id	property_id	property_alias	event_timestamp	kualitas	boolean_nilai	int_nilai	nilai_ganda	string_nilai
-	-	Li						
bf57-16180bc3d3a	b809-75287d83bcd	ne-5/ Bus-2/ Machine-5/ Temperature						
888984980b8b-42b-	3b458f00-24e7-458-b4e8-c60	/ company, windfarm turbine /7/ winddirection	157521937	BAGUS			348.75	
bf57-16180bc3d3a	26eff654a							

Note

Anda harus menyertakan klausa filter pada event_timestamp kolom untuk menanyakan raw_time_series tampilan. Ini adalah filter yang diperlukan, dan kueri akan gagal tanpanya.

Example kueri

```
SELECT event_timestamp, double_value FROM raw_time_series WHERE event_timestamp > 1234567890
```

Lihat:terbaru_Value_Time_Series

aset_id	property_id	property_alias	event_timestamp	kualitas	boolean_nilai	int_nilai	nilai_ganda	string_nilai
888984980b8b-42b1-b4e8-bf57-16180bc3d3a	3b458f00-24e7-458c-26eff654a	/company, windfarm 3/ turbine /7/ winddirection	15752196	BAGUS			355.39	

Lihat:precomputed_aggregates

aset_id	property_id	property_alias	event_timestamp	kualitas	resolusi	nilai_sum	count_value	nilai_rata-rata	maksimum_nilai	minimum_nilai	stdev_nilai
888984980b8b-42b1-b4e8-bf57-16180bc3d3a	b29be4b000-42b1-b4e8-bf57-16180bc3d3a	Roche-5/ Bus-Machine-5/ Temperature	15752196	BAGUS	15m	1105.4	15	73.4	80,6	68	3,64

Jenis data yang didukung

AWS IoT SiteWise bahasa query mendukung tipe data berikut.

Nilai skalar

Tipe data	Deskripsi
STRING	Sebuah string panjang maksimum 1024 byte.
INTEGER	Integer 32-bit yang ditandatangani dengan rentang dari. -2,147,483,648 to 2,147,483,647
DOUBLE	Nomor floating point dengan rentang dari- 10^{100} to 10^{100} , atau Nan dengan presisi IEEE 754 ganda.
BOOLEAN	true atau false.

Nullvalue: Boolean true yang menunjukkan kurangnya data yang ditentukan.

Note

Data presisi ganda tidak tepat. Beberapa nilai tidak dikonversi dengan tepat, dan tidak akan mewakili semua bilangan real karena presisi terbatas. Data floating-point dalam kueri mungkin bukan nilai yang sama yang diwakili secara internal. Nilai dibulatkan jika presisi nomor input terlalu tinggi.

Mengambil data dengan pernyataan SELECT

SELECTPernyataan ini digunakan untuk mengambil data dari satu atau lebih tampilan. AWS IoT SiteWise mendukung pandangan yang implisitJOIN. Anda dapat membuat daftar tampilan untuk bergabung (dalam FROM klausa SELECT pernyataan), menggunakan koma untuk memisahkannya.

Example

Gunakan SELECT pernyataan berikut:

```
SELECT select_expr [, ...]  
[ FROM from_item [, ...] ]
```

```
[ WHERE [LIKE condition ESCAPE condition] ]
```

Pada contoh sebelumnya, LIKE klausa menentukan kondisi pencarian dan penyaringan menggunakan kartu liar. AWS IoT SiteWise mendukung percentage (%) sebagai karakter kartu liar.

Example untuk digunakan % dalam kondisi:

```
Prefix search: String%  
Infix search: %String%  
Suffix search: %String
```

Example untuk mencari aset:

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'Wind%'
```

Example untuk mencari aset menggunakan kondisi ESCAPE:

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'room\%' ESCAPE  
'\'
```

Operator logis

AWS IoT SiteWise mendukung operator logis berikut.

Operator logis

Operator	Deskripsi	Contoh
AND	TRUE jika kedua nilai benar	a AND b

Jika salah satu a atau b adalah FALSE, ekspresi sebelumnya dievaluasi menjadi false. Agar AND operator dapat mengevaluasi menjadi benar, baik a dan b harus benar.

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
```

Operator perbandingan

AWS IoT SiteWise mendukung operator perbandingan berikut. Semua operasi perbandingan tersedia untuk tipe data bawaan dan dievaluasi ke boolean.

Operator logis

Operator	Deskripsi
<	Kurang dari
>	Lebih besar dari
<=	Kurang dari atau sama dengan
>=	Lebih besar dari atau sama dengan
=	Setara
!=	Tidak sama

Tabel kebenaran operasi perbandingan untuk nilai non numerik

Jenis	Ketik >= x	Jenis <= x	Ketik > x	Ketik < x	Jenis = x	Ketik != x
NaN	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Ada beberapa predikat yang berperilaku seperti operator tetapi memiliki sintaks khusus. Lihat di bawah untuk contoh.

Predikat perbandingan

Operator	Deskripsi
IS NULL	Menguji apakah suatu nilai adalah NULL.

Operator	Deskripsi
IS NOT NULL	Tes jika nilai tidakNULL.
IS NaN	Menguji apakah suatu nilai adalahNaN.
IS NOT NaN	Tes jika nilai tidakNaN.

Kueri contoh

Pemfilteran metadata

Contoh berikut adalah untuk pemfilteran metadata dengan SELECT pernyataan dengan bahasa query: AWS IoT SiteWise

```
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%'
```

Penyaringan nilai

Berikut ini adalah contoh pemfilteran nilai menggunakan SELECT pernyataan dengan bahasa AWS IoT SiteWise query:

```
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
```

Berinteraksi dengan AWS layanan lain

AWS IoT SiteWise dapat mempublikasikan data aset ke broker pesan berlangganan publikasi AWS IoT MQTT, sehingga Anda dapat berinteraksi dengan data aset Anda dari layanan lain. AWS IoT SiteWise menetapkan setiap properti aset topik MQTT unik yang dapat Anda gunakan untuk merutekan data aset Anda ke AWS layanan lain menggunakan aturan Inti. AWS IoT Misalnya, Anda dapat mengonfigurasi aturan AWS IoT Core untuk melakukan tugas-tugas berikut:

- Identifikasi kegagalan peralatan dan beri tahu personel yang sesuai dengan mengirimkan data ke [AWS IoT Events](#).
- Historisasi data aset tertentu untuk digunakan dalam solusi perangkat lunak eksternal dengan mengirimkan data ke Amazon [DynamoDB](#).
- Hasilkan laporan mingguan dengan memicu suatu [AWS Lambda](#) fungsi.

Anda dapat mengikuti tutorial yang berjalan melalui langkah-langkah yang diperlukan untuk mengatur aturan yang menyimpan nilai properti di DynamoDB. Untuk informasi selengkapnya, lihat [Publikasikan pembaruan nilai properti ke Amazon DynamoDB](#).

Untuk informasi selengkapnya tentang cara mengonfigurasi aturan, lihat [Aturan](#) di Panduan AWS IoT Pengembang.

Anda juga dapat menggunakan data dari AWS layanan lain kembali ke AWS IoT SiteWise. Untuk menelan data melalui tindakan AWS IoT SiteWise aturan, lihat [Menelan data untuk AWS IoT SiteWise menggunakan aturan AWS IoT Core](#).

Topik

- [Memahami properti aset dalam topik MQTT](#)
- [Aktifkan notifikasi properti aset di AWS IoT SiteWise](#)
- [Kueri pemberitahuan properti aset di AWS IoT SiteWise](#)
- [Ekspor data ke Amazon S3 dengan pemberitahuan properti aset](#)
- [Integrasikan AWS IoT SiteWise dengan Grafana](#)
- [Integrasikan AWS IoT SiteWise dan AWS IoT TwinMaker](#)
- [Deteksi anomali dengan Lookout for Equipment](#)

Memahami properti aset dalam topik MQTT

Setiap properti aset memiliki jalur topik MQTT unik dalam format berikut.

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

Note

AWS IoT SiteWise tidak mendukung wildcard filter topik # (multi-level) di mesin aturan AWS IoT Inti. Anda dapat menggunakan wildcard + (single-level). Misalnya, Anda dapat menggunakan filter topik berikut untuk mencocokkan semua pembaruan untuk model aset tertentu.

```
$aws/sitewise/asset-models/assetModelId/assets/+/properties/+
```

Untuk mempelajari lebih lanjut tentang wildcard filter topik, lihat [Topik](#) di Panduan Pengembang AWS IoT Inti.

Aktifkan notifikasi properti aset di AWS IoT SiteWise

Anda dapat mengaktifkan pemberitahuan properti untuk mempublikasikan pembaruan data aset AWS IoT Core, lalu menjalankan kueri pada data Anda. Dengan pemberitahuan properti aset, AWS IoT SiteWise berikan AWS CloudFormation templat yang dapat Anda gunakan untuk mengekspor AWS IoT SiteWise data ke Amazon S3.

Note

Data aset dikirim ke AWS IoT Core setiap kali diterima oleh AWS IoT SiteWise, terlepas dari apakah nilainya telah berubah.

Topik

- [Aktifkan notifikasi properti aset \(konsol\)](#)
- [Aktifkan notifikasi properti aset \(AWS CLI\)](#)

Aktifkan notifikasi properti aset (konsol)

Secara default, AWS IoT SiteWise tidak mempublikasikan pembaruan nilai properti. Anda dapat menggunakan AWS IoT SiteWise konsol untuk mengaktifkan notifikasi untuk properti aset.

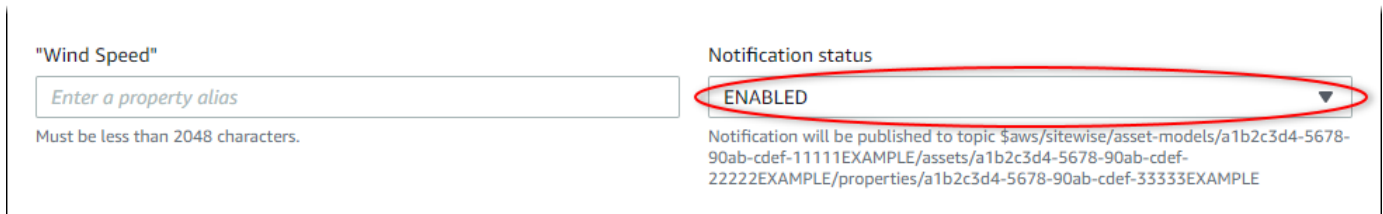
Untuk mengaktifkan atau menonaktifkan notifikasi untuk properti aset (konsol)

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset.
3. Pilih aset untuk mengaktifkan notifikasi properti.

Tip

Anda dapat memilih ikon panah untuk memperluas hierarki aset untuk menemukan aset Anda.

4. Pilih Edit.
5. Untuk status Pemberitahuan properti aset, pilih ENABLED.



The screenshot shows the AWS IoT SiteWise console interface. On the left, there is a text input field labeled '"Wind Speed"' with a placeholder text 'Enter a property alias' and a note 'Must be less than 2048 characters.' On the right, there is a 'Notification status' dropdown menu. The dropdown is open, showing 'ENABLED' as the selected option. Below the dropdown, there is a text description: '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'.

Anda juga dapat memilih DISABLED untuk menonaktifkan notifikasi untuk properti aset.

6. Pilih Simpan.

Aktifkan notifikasi properti aset (AWS CLI)

Secara default, AWS IoT SiteWise tidak mempublikasikan pembaruan nilai properti. Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk mengaktifkan atau menonaktifkan notifikasi untuk properti aset.

Anda harus mengetahui aset `assetId` dan properti Anda `propertyId` untuk menyelesaikan prosedur ini. Anda juga dapat menggunakan ID eksternal. Jika Anda membuat aset dan tidak mengetahuinya `assetId`, gunakan [ListAssets](#) API untuk mencantumkan semua aset untuk model tertentu. Gunakan [DescribeAsset](#) operasi untuk melihat properti aset Anda termasuk properti IDs.

Gunakan [UpdateAssetProperty](#) operasi untuk mengaktifkan atau menonaktifkan notifikasi untuk properti aset. Tentukan parameter berikut:

- `assetId`— ID aset.
- `propertyId`— ID properti aset.
- `propertyNotificationState`— Status pemberitahuan nilai properti: `ENABLED` atau `DISABLED`.
- `propertyAlias`— Alias properti. Tentukan alias properti yang ada saat Anda memperbarui status notifikasi. Jika Anda menghilangkan parameter ini, alias properti yang ada akan dihapus.

Untuk mengaktifkan atau menonaktifkan pemberitahuan untuk properti aset (CLI)

1. Jalankan perintah berikut untuk mengambil alias properti aset. Ganti *asset-id* dengan ID aset dan *property-id* dengan ID properti.

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

Operasi mengembalikan respons yang berisi rincian properti aset dalam format berikut. Alias properti ada `assetProperty.alias` di objek JSON.

```
{  
  "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. Jalankan perintah berikut untuk mengaktifkan notifikasi untuk properti aset. Ganti *property-alias* dengan alias properti dari respons perintah sebelumnya, atau hilangkan `--property-alias` untuk memperbarui properti tanpa alias.

```
aws iotsitewise update-asset-property \  
  --asset-id asset-id \  
  --property-id property-id \  
  --property-notification-state ENABLED \  
  --property-alias property-alias
```

Anda juga dapat meneruskan `--property-notification-state DISABLED` untuk menonaktifkan notifikasi untuk properti aset.

Kueri pemberitahuan properti aset di AWS IoT SiteWise

Untuk meminta pemberitahuan properti aset, buat AWS IoT Core aturan yang terdiri dari pernyataan SQL.

AWS IoT SiteWise menerbitkan pembaruan data properti aset ke AWS IoT Core dalam format berikut.

```
{  
  "type": "PropertyValueUpdate",  
  "payload": {  
    "assetId": "String",  
    "propertyId": "String",  
    "values": [  
      {  
        "timestamp": {  
          "timeInSeconds": Number,  
          "offsetInNanos": Number  
        },  
        "quality": "String",  
        "value": {  
          "booleanValue": Boolean,  
          "doubleValue": Number,  
          "integerValue": Number,  
          "stringValue": "String",
```

```

        "nullValue": {
            "valueType": "String"
        }
    }
}
]
}
}

```

Setiap struktur dalam `values` daftar adalah struktur timestamp-quality-value (TQV).

- `timestamp`— berisi waktu epoch Unix saat ini dalam hitungan detik dengan offset nanodetik.
- `quality`— Berisi salah satu string berikut yang menunjukkan kualitas titik data:
 - `GOOD`— Data tidak terpengaruh oleh masalah apa pun.
 - `BAD`— Data dipengaruhi oleh masalah seperti kegagalan sensor.
 - `UNCERTAIN`— Data dipengaruhi oleh masalah seperti ketidakakuratan sensor.
- `value`— Berisi salah satu bidang berikut, tergantung pada jenis properti:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

`nullValue`— Struktur dengan bidang berikut yang menunjukkan jenis nilai properti dengan nilai Null dan kualitas atau. `BAD UNCERTAIN`

- `valueType`— Enum dari {"B", "D", "S", "I"}

Untuk mengurai nilai dari `values` array, Anda perlu menggunakan kueri objek bersarang yang kompleks dalam pernyataan SQL aturan Anda. Untuk informasi selengkapnya, lihat [Kueri objek bersarang](#) di Panduan AWS IoT Pengembang, atau lihat [Publikasikan pembaruan nilai properti ke Amazon DynamoDB](#) tutorial untuk contoh spesifik penguraian pesan pemberitahuan properti aset.

Example Contoh query untuk mengekstrak array nilai

Pernyataan berikut menunjukkan bagaimana untuk query array nilai properti diperbarui untuk properti tipe ganda tertentu pada semua aset dengan properti tersebut.


```

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'

```

Pernyataan kueri aturan sebelumnya mengeluarkan data dalam format berikut.

```

{
  "windspeed": [
    26.32020195042838,
    26.282584572975477,
    26.352566977372508,
    26.283084346171442,
    26.571883739599322,
    26.60684140743005,
    26.628738636715045,
    26.273486932802125,
    26.436379105473964,
    26.600590095377303
  ]
}

```

Example Contoh kueri untuk mengekstrak satu nilai

Pernyataan berikut menunjukkan bagaimana untuk query nilai pertama dari array nilai properti untuk properti tipe ganda tertentu pada semua aset dengan properti tersebut.

```

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'

```

Pernyataan kueri aturan sebelumnya mengeluarkan data dalam format berikut.

```

{
  "windspeed": 26.32020195042838
}

```

```
}
```

Important

Pernyataan kueri aturan ini mengabaikan pembaruan nilai selain yang pertama di setiap batch. Setiap batch dapat berisi hingga 10 nilai. Jika Anda perlu menyertakan nilai yang tersisa, Anda harus menyiapkan solusi yang lebih kompleks untuk menampilkan nilai properti aset ke layanan lain. Misalnya, Anda dapat mengatur aturan dengan AWS Lambda tindakan untuk mempublikasikan ulang setiap nilai dalam larik ke topik lain, dan menyiapkan aturan lain untuk menanyakan topik tersebut dan mempublikasikan setiap nilai ke tindakan aturan yang diinginkan.

Ekspor data ke Amazon S3 dengan pemberitahuan properti aset

Anda dapat mengekspor data masuk dari AWS IoT SiteWise ke bucket Amazon S3 di akun Anda. Anda dapat mencadangkan data Anda dalam format yang dapat Anda gunakan untuk membuat laporan historis atau untuk menganalisis data Anda dengan metode yang kompleks.

Untuk mengekspor data deret waktu dari AWS IoT SiteWise, aktifkan fitur tingkat dingin agar data disimpan di bucket Amazon S3. Lihat [Mengelola penyimpanan data AWS IoT SiteWise](#) untuk detailnya.

Untuk mengekspor model aset dan metadata aset dari AWS IoT SiteWise, gunakan fitur operasi massal untuk mengekspor metadata ke bucket Amazon S3. Lihat [Operasi massal dengan aset dan model](#) untuk detailnya.

Integrasikan AWS IoT SiteWise dengan Grafana

Grafana adalah platform visualisasi data yang digunakan untuk memvisualisasikan dan memantau data di dasbor. Di Grafana versi 10.4.0 dan yang lebih baru, gunakan AWS IoT SiteWise plugin untuk memvisualisasikan AWS IoT SiteWise data aset Anda di dasbor Grafana. Pengguna dapat memvisualisasikan data dari berbagai AWS sumber (seperti AWS IoT SiteWise, Amazon Timestream, dan CloudWatch Amazon) dan sumber data lainnya dengan satu dasbor Grafana.

Anda memiliki dua opsi untuk menggunakan AWS IoT SiteWise plugin:

- Server Grafana lokal

Anda dapat mengatur AWS IoT SiteWise plugin di server Grafana yang Anda kelola. Untuk informasi lebih lanjut tentang cara menambahkan dan menggunakan plugin, lihat file [AWS IoT SiteWise Datasource README](#) di situs web. GitHub

- AWS Managed Service for Grafana

Anda dapat menggunakan AWS IoT SiteWise plugin di AWS Managed Service for Grafana (AMG). AMG mengelola server Grafana untuk Anda sehingga Anda dapat memvisualisasikan data Anda tanpa harus membangun, mengemas, atau menyebarkan perangkat keras apa pun atau infrastruktur Grafana lainnya. Untuk informasi selengkapnya, lihat topik berikut di AWS Managed Service for Grafana User Guide:

- [Apa itu Amazon Managed Service for Grafana \(AMG\)?](#)
- [Menggunakan sumber AWS IoT SiteWise data](#)

Example Contoh dasbor Grafana

Dasbor Grafana berikut memvisualisasikan [demo](#) ladang angin. Anda dapat mengakses dasbor demo ini di situs web [Grafana Play](#).



Integrasikan AWS IoT SiteWise dan AWS IoT TwinMaker

Mengintegrasikan dengan AWS IoT TwinMaker memberikan akses ke fungsionalitas yang kuat di AWS IoT SiteWise, seperti ExecuteQuery API pengambilan AWS IoT SiteWise data dan pencarian aset lanjutan di konsol. AWS IoT SiteWise Untuk mengintegrasikan layanan dan menggunakan fitur-fitur ini, Anda harus terlebih dahulu mengaktifkan integrasi.

Topik

- [Mengaktifkan integrasi](#)
- [Integrasi dan AWS IoT SiteWiseAWS IoT TwinMaker](#)

Mengaktifkan integrasi

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa. Elemen Action dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Untuk informasi selengkapnya tentang tindakan yang AWS IoT SiteWise didukung, lihat [Tindakan yang ditentukan oleh AWS IoT SiteWise](#) dalam Referensi Otorisasi Layanan.

Untuk informasi selengkapnya tentang peran AWS IoT TwinMaker terkait layanan, lihat Peran [terkait layanan AWS IoT TwinMaker di Panduan Pengguna](#).AWS IoT TwinMaker

Sebelum Anda dapat mengintegrasikan AWS IoT SiteWise dan AWS IoT TwinMaker, Anda harus memberikan izin berikut yang memungkinkan AWS IoT SiteWise untuk mengintegrasikan dengan ruang kerja AWS IoT TwinMaker tertaut:

- `iotsitewise:EnableSiteWiseIntegration`— Memungkinkan AWS IoT SiteWise untuk berintegrasi dengan AWS IoT TwinMaker ruang kerja yang ditautkan. Integrasi ini memungkinkan AWS IoT TwinMaker untuk membaca semua informasi pemodelan Anda AWS IoT SiteWise melalui peran AWS IoT TwinMaker terkait layanan. Untuk mengaktifkan izin ini, tambahkan kebijakan berikut ke peran IAM Anda:

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

Integrasi dan AWS IoT SiteWiseAWS IoT TwinMaker

Untuk mengintegrasikan AWS IoT SiteWise dan AWS IoT TwinMaker, Anda harus memiliki yang berikut:

- AWS IoT SiteWise pengaturan peran terkait layanan di akun Anda
- AWS IoT TwinMaker pengaturan peran terkait layanan di akun Anda
- AWS IoT TwinMaker ruang kerja dengan ID `IoTSiteWiseDefaultWorkspace` di akun Anda di Wilayah.

Untuk mengintegrasikan dengan menggunakan AWS IoT SiteWise konsol

Saat Anda melihat AWS IoT TwinMaker spanduk Integrasi dengan di konsol, pilih Berikan izin. Prasyarat dibuat di akun Anda.

Untuk mengintegrasikan dengan menggunakan AWS CLI

Untuk mengintegrasikan AWS IoT SiteWise dan AWS IoT TwinMaker dengan menggunakan AWS CLI, masukkan perintah berikut:

1. Panggil `CreateServiceLinkedRole` dengan sebuah `AWSServiceName` dari `iotsitewise.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com
```

2. Panggil `CreateServiceLinkedRole` dengan sebuah `AWSServiceName` dari `iottwinmaker.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iottwinmaker.amazonaws.com
```

3. Panggil `CreateWorkspace` dengan sebuah ID dari `IoTSiteWiseDefaultWorkspace`.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

Deteksi anomali dengan Lookout for Equipment

Note

Deteksi anomali hanya tersedia di Wilayah di mana Amazon Lookout for Equipment tersedia.

Anda dapat berintegrasi AWS IoT SiteWise dengan Amazon Lookout for Equipment untuk mendapatkan wawasan tentang peralatan industri Anda melalui deteksi anomali dan pemeliharaan

prediktif peralatan industri. Lookout for Equipment adalah layanan machine learning (ML) untuk memantau peralatan industri yang mendeteksi perilaku peralatan abnormal dan mengidentifikasi potensi kegagalan. Dengan Lookout for Equipment, Anda dapat menerapkan program pemeliharaan prediktif dan mengidentifikasi proses peralatan yang kurang optimal. Untuk informasi selengkapnya tentang Lookout for Equipment, lihat [Apa itu Amazon Lookout for Equipment?](#) di Panduan Pengguna Amazon Lookout for Equipment.

Saat Anda membuat prediksi untuk melatih model ML guna mendeteksi perilaku peralatan anomali, AWS IoT SiteWise kirimkan nilai properti aset ke Lookout for Equipment untuk melatih model ML guna mendeteksi perilaku peralatan anomali. Untuk menentukan definisi prediksi pada model aset, Anda menentukan peran IAM yang diperlukan untuk Lookout for Equipment untuk mengakses data dan properti yang akan dikirim ke Lookout for Equipment dan mengirim data yang diproses ke Amazon S3. Untuk informasi selengkapnya, lihat [Buat model aset di AWS IoT SiteWise](#).

Untuk mengintegrasikan AWS IoT SiteWise dan Lookout for Equipment, Anda akan melakukan langkah-langkah tingkat tinggi berikut:

- Tambahkan definisi prediksi pada model aset yang menguraikan properti apa yang ingin Anda lacak. Definisi prediksi adalah kumpulan pengukuran, transformasi, dan metrik yang dapat digunakan kembali yang digunakan untuk membuat prediksi pada aset yang didasarkan pada model aset tersebut.
- Latih prediksi berdasarkan data historis yang Anda berikan.
- Jadwal inferensi, yang menceritakan AWS IoT SiteWise seberapa sering menjalankan prediksi tertentu.

Setelah inferensi dijadwalkan, model Lookout for Equipment memantau data yang diterimanya dari peralatan Anda dan mencari anomali dalam perilaku peralatan. Anda dapat melihat dan menganalisis hasil di SiteWise Monitor, menggunakan operasi AWS IoT SiteWise GET API, atau konsol Lookout for Equipment. Anda juga dapat membuat alarm menggunakan detektor alarm dari model aset untuk mengingatkan Anda tentang perilaku peralatan yang tidak normal.

Topik

- [Tambahkan definisi prediksi \(konsol\)](#)
- [Latih prediksi \(konsol\)](#)
- [Memulai atau menghentikan inferensi pada prediksi \(konsol\)](#)
- [Tambahkan definisi prediksi \(CLI\)](#)

- [Latih prediksi dan inferensi awal \(CLI\)](#)
- [Melatih prediksi \(CLI\)](#)
- [Memulai atau menghentikan inferensi pada prediksi \(CLI\)](#)

Tambahkan definisi prediksi (konsol)

Untuk mulai mengirim data yang dikumpulkan oleh AWS IoT SiteWise Lookout for Equipment, Anda harus menambahkan definisi prediksi ke model AWS IoT SiteWise aset.

Untuk menambahkan definisi prediksi ke model AWS IoT SiteWise aset

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Model dan pilih model aset yang ingin Anda tambahkan definisi prediksi.
3. Pilih Prediksi.
4. Pilih Tambahkan definisi prediksi.
5. Tentukan detail tentang definisi prediksi.
 - a. Masukkan Nama unik dan Deskripsi untuk definisi prediksi Anda. Pilih nama dengan serius karena setelah Anda membuat definisi prediksi, Anda tidak dapat mengubah namanya.
 - b. Buat atau pilih peran izin IAM yang memungkinkan Anda membagikan data aset Anda AWS IoT SiteWise dengan Amazon Lookout for Equipment. Peran tersebut harus memiliki IAM dan kebijakan kepercayaan berikut. Untuk bantuan membuat peran, lihat [Membuat peran menggunakan kebijakan kepercayaan khusus \(konsol\)](#).

Kebijakan IAM

```
{
  "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"
}
]

```

```
}

```

Kebijakan kepercayaan

```
{
  "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. Pilih Berikutnya.

6. Pilih atribut data (pengukuran, transformasi, dan metrik) yang ingin Anda kirim ke Lookout for Equipment.
 - a. (Opsional) Pilih pengukuran.
 - b. (Opsional) Pilih transformasi.
 - c. (Opsional) Pilih metrik.
 - d. Pilih Berikutnya.
7. Tinjau pilihan Anda. Untuk menambahkan definisi prediksi ke model aset, pada halaman ringkasan, pilih Tambahkan definisi prediksi.

Anda juga dapat Mengedit atau Menghapus definisi prediksi yang ada yang memiliki prediksi aktif yang dilampirkan.

Latih prediksi (konsol)

Setelah Anda menambahkan definisi prediksi ke model aset, Anda dapat melatih prediksi yang ada di aset Anda.

Untuk melatih prediksi di AWS IoT SiteWise

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset, dan pilih aset yang ingin Anda pantau.
3. Pilih Prediksi.
4. Pilih prediksi yang ingin Anda latih.
5. Di bawah Tindakan, pilih Mulai pelatihan, dan lakukan hal berikut:
 - a. Di bawah Detail Prediksi, pilih peran izin IAM yang memungkinkan AWS IoT SiteWise untuk berbagi data aset Anda dengan Lookout for Equipment. Jika Anda perlu membuat peran baru, pilih Buat peran baru.
 - b. Untuk pengaturan data Pelatihan, masukkan rentang waktu data Pelatihan untuk memilih data mana yang akan digunakan untuk melatih prediksi.
 - c. (Opsional) Pilih laju pengambilan sampel data setelah pemrosesan pasca.
 - d. (Opsional) Untuk label Data, berikan bucket dan awalan Amazon S3 yang menyimpan data pelabelan Anda. Untuk informasi selengkapnya tentang pelabelan data, lihat [Pelabelan data Anda di Panduan Pengguna](#) Amazon Lookout for Equipment.
 - e. Pilih Berikutnya.

6. (Opsional) Jika Anda ingin prediksi aktif segera setelah menyelesaikan pelatihan, di bawah Pengaturan lanjutan, pilih Aktifkan prediksi secara otomatis setelah pelatihan, lalu lakukan hal berikut:
 - a. Di bawah Input data, untuk frekuensi upload Data, tentukan seberapa sering data diunggah, dan untuk waktu tunda Offset, tentukan berapa banyak buffer yang akan digunakan.
 - b. Pilih Berikutnya.
7. Tinjau detail prediksi dan pilih Simpan dan mulai.

Memulai atau menghentikan inferensi pada prediksi (konsol)

Note

Biaya Lookout for Equipment berlaku untuk kesimpulan terjadwal dengan data yang ditransfer antara dan AWS IoT SiteWise Lookout for Equipment. Untuk informasi selengkapnya, lihat [harga Amazon Lookout](#) for Equipment.

Jika Anda menambahkan prediksi `lookoutequipment:CreateDataset`, tetapi tidak memilih untuk mengaktifkannya setelah pelatihan, Anda harus mengaktifkannya untuk mulai memantau aset Anda.

Untuk memulai inferensi untuk prediksi

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Aset, dan pilih aset tempat prediksi ditambahkan.
3. Pilih Prediksi.
4. Pilih prediksi yang ingin Anda aktifkan.
5. Di bawah Tindakan, pilih Mulai inferensi, dan lakukan hal berikut:
 - a. Di bawah Input data, untuk frekuensi upload Data, tentukan seberapa sering data diunggah, dan untuk waktu tunda Offset, tentukan berapa banyak buffer yang akan digunakan.
 - b. Pilih Simpan dan mulai.

Untuk menghentikan inferensi untuk prediksi

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di panel navigasi, pilih Aset, dan pilih aset tempat prediksi ditambahkan.
3. Pilih Prediksi.
4. Pilih prediksi yang ingin Anda hentikan.
5. Di bawah Tindakan, pilih Hentikan inferensi.

Tambahkan definisi prediksi (CLI)

Untuk menentukan definisi prediksi pada model aset baru atau yang sudah ada, Anda dapat menggunakan AWS Command Line Interface (AWS CLI). Setelah Anda menentukan definisi prediksi pada model aset, Anda melatih, dan menjadwalkan inferensi untuk, prediksi pada aset AWS IoT SiteWise untuk melakukan deteksi anomali dengan Lookout for Equipment.

Prasyarat

Untuk menyelesaikan langkah-langkah ini, Anda harus memiliki model aset dan setidaknya satu aset yang dibuat. Untuk informasi selengkapnya, silakan lihat [Buat model aset \(AWS CLI\)](#) dan [Buat aset \(AWS CLI\)](#).

Jika Anda baru AWS IoT SiteWise, Anda harus memanggil operasi `CreateBulkImportJob` API untuk mengimpor nilai properti aset ke dalam AWS IoT SiteWise, yang akan digunakan untuk melatih model. Untuk informasi selengkapnya, lihat [Buat pekerjaan impor AWS IoT SiteWise massal \(AWS CLI\)](#).

Untuk menambahkan definisi prediksi

1. Buat file bernama `asset-model-payload.json`. Ikuti langkah-langkah di bagian lain ini untuk menambahkan detail model aset Anda ke file, tetapi jangan kirimkan permintaan untuk membuat atau memperbarui model aset.
 - Untuk informasi selengkapnya tentang cara membuat model aset, lihat [Buat model aset \(AWS CLI\)](#)
 - Untuk informasi selengkapnya tentang cara memperbarui model aset yang ada, lihat [Memperbarui model aset atau komponen \(AWS CLI\)](#)
2. Tambahkan Lookout for Equipment composite model `assetModelCompositeModels()` ke model aset dengan menambahkan kode berikut.
 - Ganti *Property* dengan ID properti yang ingin Anda sertakan. Untuk mendapatkan itu IDs, hubungi [DescribeAssetModel](#).

- Ganti *RoleARN* dengan ARN dari peran IAM yang memungkinkan Lookout for Equipment mengakses data Anda. 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",

```

```

        "dataKeySpec": "AWS/L4E_ANOMALY_DATASET",
        "type": {
            "attribute": {}
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_MODEL",
        "dataType": "STRUCT",
        "dataKeySpec": "AWS/L4E_ANOMALY_MODEL",
        "type": {
            "attribute": {}
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_INFERENCE",
        "dataType": "STRUCT",
        "dataKeySpec": "AWS/L4E_ANOMALY_INFERENCE",
        "type": {
            "attribute": {}
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_TRAINING_STATUS",
        "dataType": "STRUCT",
        "dataKeySpec": "AWS/L4E_ANOMALY_TRAINING_STATUS",
        "type": {
            "attribute": {
                "defaultValue": "{}"
            }
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
        "dataType": "STRUCT",
        "dataKeySpec": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
        "type": {
            "attribute": {
                "defaultValue": "{}"
            }
        }
    }
]
}

```

3. Buat model aset atau perbarui model aset yang ada. Lakukan salah satu hal berikut ini:

- Untuk membuat model aset, jalankan perintah berikut:

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Untuk memperbarui model aset yang ada, jalankan perintah berikut. Ganti *asset-model-id* dengan ID model aset yang ingin Anda perbarui.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Setelah Anda menjalankan perintah, perhatikan `assetModelId` dalam respon.

Latih prediksi dan inferensi awal (CLI)

Sekarang setelah definisi prediksi ditentukan, Anda dapat melatih aset berdasarkan itu dan memulai inferensi. Jika Anda ingin melatih prediksi Anda tetapi tidak memulai inferensi, lompat ke [Melatih prediksi \(CLI\)](#). Untuk melatih prediksi dan memulai inferensi pada aset, Anda memerlukan sumber `assetId` daya target.

Untuk melatih dan memulai inferensi prediksi

1. Jalankan perintah berikut untuk menemukan bagian `assetModelCompositeModelId` bawah `assetModelCompositeModelSummaries`. Ganti *asset-model-id* dengan ID model aset yang Anda buat [Memperbarui model aset atau komponen \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Jalankan perintah berikut untuk menemukan `TrainingWithInference` tindakan. `actionDefinitionId` Ganti *asset-model-id* dengan ID yang digunakan pada langkah sebelumnya dan ganti *asset-model-composite-model-id* dengan ID yang dikembalikan pada langkah sebelumnya.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
```



```
--asset-model-composite-model-id asset-model-composite-model-id \
```

3. Buat file bernama `train-start-inference-prediction.json` dan tambahkan kode berikut, ganti yang berikut ini:

- *asset-id* dengan ID aset target
- *action-definition-id* dengan ID TrainingWithInference tindakan
- *StartTime* dengan dimulainya data pelatihan, disediakan dalam hitungan detik
- *EndTime* dengan berakhirnya data pelatihan, disediakan dalam hitungan detik
- *TargetSamplingRate* dengan laju pengambilan sampel data setelah pemrosesan pasca oleh Lookout for Equipment. Nilai yang diizinkan adalah: 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. Jalankan perintah berikut untuk memulai pelatihan dan inferensi:

```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-prediction.json
```

Melatih prediksi (CLI)

Sekarang setelah definisi prediksi ditentukan, Anda dapat melatih aset berdasarkan itu. Untuk melatih prediksi aset, Anda memerlukan sumber daya target. `assetId`

Untuk melatih prediksi

1. Jalankan perintah berikut untuk menemukan bagian `assetModelCompositeModelId` bawah `assetModelCompositeModelSummaries`. Ganti *asset-model-id* dengan ID model aset yang Anda buat [Memperbarui model aset atau komponen \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \
```

2. Jalankan perintah berikut untuk menemukan Training tindakan. `actionDefinitionId` Ganti *asset-model-id* dengan ID yang digunakan pada langkah sebelumnya dan ganti *asset-model-composite-model-id* dengan ID yang dikembalikan pada langkah sebelumnya.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Buat file bernama `train-prediction.json` dan tambahkan kode berikut, ganti yang berikut ini:

- *asset-id* dengan ID aset target
- *action-definition-id* dengan ID tindakan pelatihan
- *StartTime* dengan dimulainya data pelatihan, disediakan dalam hitungan detik
- *EndTime* dengan berakhirnya data pelatihan, disediakan dalam hitungan detik
- (Opsional) *BucketName* dengan nama bucket Amazon S3 yang menyimpan data label Anda
- (Opsional) *Prefix* dengan awalan yang terkait dengan bucket Amazon S3.
- *TargetSamplingRate* dengan laju pengambilan sampel data setelah pemrosesan pasca oleh Lookout for Equipment. Nilai yang diizinkan adalah: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

Note

Sertakan nama bucket dan awalan atau keduanya.

```
{  
  "targetResource": {  
    "assetId": "asset-id"
```

```

    },
    "actionDefinitionId": "action-definition-Id",
    "actionPayload":{ "stringValue": "{\"l4ETraining\": {\"trainingMode\":
\"START\", \"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime,
\"targetSamplingRate\": \"TargetSamplingRate\", \"labelInputConfiguration\":
{ \"bucketName\": \"BucketName\", \"prefix\": \"Prefix\"} } }"}
  }
}

```

4. Jalankan perintah berikut untuk memulai pelatihan:

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

Sebelum Anda dapat memulai inferensi, pelatihan harus diselesaikan. Untuk memeriksa status pelatihan, lakukan salah satu hal berikut:

- Dari konsol, arahkan ke aset tempat prediksi aktif.
- Dari AWS CLI, hubungi BatchGetAssetPropertyValue menggunakan trainingStatus properti. propertyId

Memulai atau menghentikan inferensi pada prediksi (CLI)

Setelah prediksi dilatih, Anda dapat memulai inferensi untuk memberi tahu Lookout for Equipment untuk mulai memantau aset Anda. Untuk memulai atau menghentikan inferensi, Anda memerlukan assetId sumber daya target.

Untuk memulai inferensi

1. Jalankan perintah berikut untuk menemukan bagian assetModelCompositeModelId bawah assetModelCompositeModelSummaries. Ganti *asset-model-id* dengan ID model aset yang Anda buat [Memperbarui model aset atau komponen \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Jalankan perintah berikut untuk menemukan Inference tindakan. actionDefinitionId Ganti *asset-model-id* dengan ID yang digunakan pada langkah sebelumnya dan ganti *asset-model-composite-model-id* dengan ID yang dikembalikan pada langkah sebelumnya.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Buat file bernama `start-inference.json` dan tambahkan kode berikut, ganti yang berikut ini:

- *asset-id* dengan ID aset target
- *action-definition-id* dengan ID dari tindakan inferensi awal
- *Offset* dengan jumlah buffer yang akan digunakan
- *Frequency* dengan seberapa sering data diunggah

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-id",
  "actionPayload": { "stringValue": "{\"l4EInference\": {\"inferenceMode\": \"START\", \"dataDelayOffsetInMinutes\": Offset, \"dataUploadFrequency\": \"Frequency\"}}"}
}
```

4. Jalankan perintah berikut untuk memulai inferensi:

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```

Untuk menghentikan inferensi

1. Jalankan perintah berikut untuk menemukan bagian `assetModelCompositeModelId` bawah `assetModelCompositeModelSummaries`. Ganti *asset-model-id* dengan ID model aset yang Anda buat [Memperbarui model aset atau komponen \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Jalankan perintah berikut untuk menemukan Inference tindakan. `actionDefinitionId`
Ganti *asset-model-id* dengan ID yang digunakan pada langkah sebelumnya dan ganti *asset-model-composite-model-id* dengan ID yang dikembalikan pada langkah sebelumnya.

```
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://stop-inference.json
```

3. Buat file bernama `stop-inference.json` dan tambahkan kode berikut, ganti yang berikut ini:

- *asset-id* dengan ID aset target
- *action-definition-id* dengan ID dari tindakan inferensi awal

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-id",  
  "actionPayload": { "stringValue": "{\"l4EIInference\":{\"inferenceMode\":\"STOP\"}}"}  
}
```

4. Jalankan perintah berikut untuk menghentikan inferensi:

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

Mengelola penyimpanan data di AWS IoT SiteWise

Anda dapat mengonfigurasi AWS IoT SiteWise untuk menyimpan data Anda di tingkatan penyimpanan berikut:

Tingkat panas

Tingkat penyimpanan panas adalah penyimpanan deret waktu yang AWS IoT SiteWise dikelola. Hot tier paling efektif untuk data yang sering diakses, dengan write-to-read latensi rendah. Data yang disimpan di hot tier digunakan oleh aplikasi industri yang membutuhkan akses cepat ke nilai pengukuran terbaru di peralatan Anda. Ini termasuk aplikasi yang memvisualisasikan metrik real-time dengan dasbor interaktif, atau aplikasi yang memantau operasi dan meluncurkan alarm untuk mengidentifikasi masalah kinerja.

Secara default, data yang tertelan AWS IoT SiteWise disimpan di hot tier. Anda dapat menentukan periode retensi untuk tingkat panas, setelah itu AWS IoT SiteWise memindahkan data di tingkat panas ke penyimpanan tingkat hangat atau dingin, berdasarkan konfigurasi Anda. Untuk kinerja terbaik dan efisiensi biaya, atur periode retensi hot tier Anda menjadi lebih lama dari waktu yang dibutuhkan untuk sering mengambil data. Ini digunakan untuk metrik waktu nyata, alarm, dan skenario pemantauan. Jika periode retensi tidak ditetapkan, data Anda disimpan tanpa batas waktu di tingkat panas.

Tingkat hangat

Tingkat penyimpanan hangat adalah tingkat AWS IoT SiteWise terkelola yang efektif untuk penyimpanan data historis yang hemat biaya. Ini paling baik digunakan untuk mengambil volume besar data dengan karakteristik write-to-read latensi sedang. Gunakan tingkat hangat untuk menyimpan data historis yang diperlukan untuk beban kerja yang besar. Misalnya, ini digunakan untuk pengambilan data untuk analitik, aplikasi intelijen bisnis (BI), alat pelaporan, dan pelatihan model pembelajaran mesin (ML). Jika Anda mengaktifkan tingkat penyimpanan dingin, Anda dapat menentukan periode retensi tingkat hangat. Setelah periode retensi berakhir, AWS IoT SiteWise menghapus data dari tingkat hangat.

Tingkat dingin

Tingkat penyimpanan dingin menggunakan bucket Amazon S3 untuk menyimpan data yang jarang digunakan. Dengan tingkat dingin diaktifkan, AWS IoT SiteWise mereplikasi deret waktu, termasuk pengukuran, metrik, transformasi dan agregat, dan definisi model aset setiap 6 jam. Tingkat dingin digunakan untuk menyimpan data yang mentolerir latensi baca tinggi untuk laporan historis dan cadangan.

Topik

- [Konfigurasi pengaturan penyimpanan di AWS IoT SiteWise](#)
- [Memecahkan masalah pengaturan penyimpanan untuk AWS IoT SiteWise](#)
- [Jalur file dan skema data yang disimpan di tingkat dingin](#)

Konfigurasi pengaturan penyimpanan di AWS IoT SiteWise

Anda dapat mengonfigurasi pengaturan penyimpanan untuk ikut serta dalam layanan penyimpanan tingkat hangat yang dikelola, dan juga untuk mereplikasi data ke tingkat dingin. Untuk mempelajari lebih lanjut tentang periode retensi untuk tingkat hangat dan panas, lihat [Dampak retensi data](#). Saat mengonfigurasi pengaturan penyimpanan, lakukan hal berikut:

- Retensi tingkat panas — Tetapkan periode retensi untuk berapa lama data Anda disimpan di hot tier sebelum dihapus, dan dipindahkan ke layanan penyimpanan tingkat hangat atau penyimpanan tingkat dingin yang dikelola berdasarkan pengaturan penyimpanan Anda. AWS IoT SiteWise akan menghapus data apa pun di hot tier yang ada sebelum periode retensi berakhir. Jika Anda tidak menetapkan periode retensi, data Anda disimpan tanpa batas waktu di tingkat panas.
- Retensi tingkat hangat — Tetapkan periode retensi untuk berapa lama data Anda disimpan di tingkat hangat sebelum dihapus dari AWS IoT SiteWise penyimpanan dan dipindahkan ke penyimpanan tingkat dingin yang dikelola pelanggan. AWS IoT SiteWise menghapus data apa pun dari tingkat hangat yang ada sebelum periode retensi berakhir. Jika periode retensi tidak diatur, data Anda disimpan tanpa batas waktu di tingkat hangat.

Note

Untuk meningkatkan kinerja kueri, tetapkan periode retensi tingkat panas dengan penyimpanan tingkat hangat.

Dampak retensi data dalam penyimpanan tingkat panas dan hangat

- Saat Anda mengurangi periode retensi penyimpanan tingkat panas, data dipindahkan secara permanen dari tingkat panas ke tingkat hangat atau dingin. Saat Anda mengurangi periode retensi tingkat hangat, data dipindahkan ke tingkat dingin, dan dihapus secara permanen dari tingkat hangat.

- Saat Anda meningkatkan periode retensi penyimpanan tingkat panas atau hangat, perubahan memengaruhi data yang dikirim AWS IoT SiteWise sejak saat itu. AWS IoT SiteWise tidak mengambil data dari penyimpanan hangat atau dingin untuk mengisi tingkat panas. Misalnya, jika periode retensi penyimpanan hot tier awalnya ditetapkan selama 30 hari dan kemudian ditingkatkan menjadi 60 hari, dibutuhkan 30 hari untuk penyimpanan hot tier berisi data senilai 60 hari.

Topik

- [Konfigurasi pengaturan penyimpanan untuk tingkat hangat \(konsol\)](#)
- [Konfigurasi pengaturan penyimpanan untuk tingkat hangat \(AWS CLI\)](#)
- [Konfigurasi pengaturan penyimpanan untuk tingkat dingin \(konsol\)](#)
- [Konfigurasi pengaturan penyimpanan untuk tingkat dingin \(AWS CLI\)](#)

Konfigurasi pengaturan penyimpanan untuk tingkat hangat (konsol)

Prosedur berikut menunjukkan cara mengonfigurasi pengaturan penyimpanan untuk mereplikasi data ke tingkat hangat di AWS IoT SiteWise konsol.

Untuk mengonfigurasi pengaturan penyimpanan di konsol

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, di bawah Pengaturan, pilih Penyimpanan.
3. Di sudut kanan atas, pilih Edit.
4. Pada halaman Edit penyimpanan, lakukan hal berikut:
5. Untuk pengaturan Hot tier, lakukan hal berikut:
 - Jika Anda ingin menetapkan periode retensi untuk berapa lama data disimpan di hot tier sebelum dihapus, dan dipindahkan ke penyimpanan tingkat hangat yang dikelola layanan, pilih Aktifkan periode retensi.
 - Untuk mengonfigurasi periode retensi, masukkan bilangan bulat dan pilih unit. Periode retensi harus lebih besar dari atau sama dengan 30 hari.

AWS IoT SiteWise menghapus data apa pun di tingkat panas yang lebih lama dari periode retensi. Jika Anda tidak menetapkan periode retensi, data Anda disimpan tanpa batas waktu.

6. (Disarankan) Untuk pengaturan tingkat hangat, lakukan hal berikut:

- Untuk memilih penyimpanan tingkat hangat, pilih Saya mengonfirmasi keikutsertaan penyimpanan tingkat hangat untuk memilih penyimpanan tingkat hangat.
- (Opsional) Untuk mengonfigurasi periode retensi, masukkan bilangan bulat dan pilih unit. Periode retensi harus lebih besar dari atau sama dengan 365 hari.

AWS IoT SiteWise menghapus data di tingkat hangat yang ada lebih awal dari periode retensi. Jika Anda tidak menetapkan periode retensi, data Anda disimpan tanpa batas waktu.

Note

- Saat Anda memilih tingkat hangat, konfigurasi hanya akan ditampilkan sekali.
- Untuk mengatur retensi tingkat panas, Anda harus memiliki penyimpanan tingkat hangat atau dingin. Untuk efisiensi biaya dan pengambilan data historis, AWS IoT SiteWise merekomendasikan agar Anda menyimpan data jangka panjang di tingkat hangat.
- Untuk mengatur retensi tingkat hangat, Anda harus memiliki penyimpanan tingkat dingin.

7. Pilih Simpan untuk menyimpan pengaturan penyimpanan Anda.

Di bagian AWS IoT SiteWise penyimpanan, penyimpanan tingkat hangat berada di salah satu status berikut:

- Diaktifkan — Jika data Anda ada sebelum periode retensi tingkat panas AWS IoT SiteWise , pindahkan data ke tingkat hangat.”
- Dinonaktifkan - Penyimpanan tingkat hangat dinonaktifkan.

Konfigurasi pengaturan penyimpanan untuk tingkat hangat (AWS CLI)

Anda dapat mengonfigurasi pengaturan penyimpanan untuk memindahkan data ke tingkat hangat dengan menggunakan perintah AWS CLI dan perintah berikut.

Untuk mencegah penggantian konfigurasi yang ada, ambil informasi konfigurasi penyimpanan saat ini dengan menjalankan perintah berikut:

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

Example respons tanpa konfigurasi tingkat dingin yang ada

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Example respons dengan konfigurasi tingkat dingin yang ada

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/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"
}
```

Konfigurasi pengaturan penyimpanan untuk tingkat hangat dengan AWS CLI

Jalankan perintah berikut untuk mengkonfigurasi pengaturan penyimpanan. Ganti file-name dengan nama file yang berisi konfigurasi AWS IoT SiteWise penyimpanan.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise konfigurasi dengan tingkat panas dan hangat

```
{
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "disassociatedDataStorage": "ENABLED",
    "warmTier": "ENABLED",
    "retentionPeriod": {
        "numberOfDays": hot-tier-retention-in-days
    }
}
```

hot-tier-retention-in-days harus bilangan bulat yang lebih besar dari atau sama dengan 30 hari.

Example response

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

Jika Anda mengaktifkan penyimpanan tingkat dingin, lihat [Konfigurasi pengaturan penyimpanan dengan AWS CLI dan tingkat dingin yang ada](#).

Konfigurasi pengaturan penyimpanan dengan AWS CLI dan tingkat dingin yang ada

Konfigurasi pengaturan penyimpanan menggunakan AWS CLI penyimpanan tingkat dingin yang ada

- Jalankan perintah berikut untuk mengkonfigurasi pengaturan penyimpanan. Ganti *file-name* dengan nama file yang berisi konfigurasi AWS IoT SiteWise penyimpanan.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise konfigurasi penyimpanan

- Ganti *amzn-s3-demo-bucket* dengan nama bucket Amazon S3 Anda.

- Ganti *prefix* dengan awalan Amazon S3 Anda.
- Ganti *aws-account-id* dengan ID AWS akun Anda.
- Ganti *role-name* dengan nama peran akses Amazon S3 yang memungkinkan AWS IoT SiteWise untuk mengirim data ke Amazon S3.
- Ganti *hot-tier-retention-in-days* dengan bilangan bulat yang lebih besar dari atau sama dengan 30 hari.
- Ganti *warm-tier-retention-in-days* dengan bilangan bulat yang lebih besar dari atau sama dengan 365 hari.

 Note

AWS IoT SiteWise akan menghapus data apa pun di tingkat hangat yang lebih tua dari periode retensi tingkat dingin. Jika Anda tidak menetapkan periode retensi, data Anda disimpan tanpa batas waktu.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/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 response

```
{
  "storageType": "MULTI_LAYER_STORAGE",
```

```
"configurationStatus": {  
  "state": "UPDATE_IN_PROGRESS"  
}  
}
```

Konfigurasi pengaturan penyimpanan untuk tingkat dingin (konsol)

Prosedur berikut menunjukkan cara mengonfigurasi pengaturan penyimpanan untuk mereplikasi data ke tingkat dingin di AWS IoT SiteWise konsol.

Untuk mengonfigurasi pengaturan penyimpanan di konsol

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, di bawah Pengaturan, pilih Penyimpanan.
3. Di sudut kanan atas, pilih Edit.
4. Pada halaman Edit penyimpanan, lakukan hal berikut:
 - a. Untuk pengaturan Penyimpanan, pilih Aktifkan penyimpanan tingkat dingin. Penyimpanan tingkat dingin dinonaktifkan secara default.
 - b. Untuk lokasi bucket S3, masukkan nama bucket Amazon S3 yang ada dan awalan.

Note

- Amazon S3 menggunakan awalan sebagai nama folder di bucket Amazon S3. Awalan harus memiliki 1-255 karakter dan diakhiri dengan garis miring (/). AWS IoT SiteWise Data Anda disimpan di folder ini.
- Jika Anda tidak memiliki bucket Amazon S3, pilih View, lalu buat bucket di konsol Amazon S3. Untuk informasi selengkapnya, lihat [Membuat bucket S3 pertama Anda](#) di Panduan Pengguna Amazon S3.

- c. Untuk peran akses S3, lakukan salah satu hal berikut:
 - Pilih Buat peran dari templat AWS terkelola, AWS secara otomatis membuat peran IAM yang memungkinkan AWS IoT SiteWise untuk mengirim data ke Amazon S3.
 - Pilih Gunakan peran yang ada, lalu pilih peran yang Anda buat dari daftar.

Note

- Anda harus menggunakan nama bucket Amazon S3 yang sama untuk lokasi bucket S3 yang Anda gunakan pada langkah sebelumnya dan dalam kebijakan IAM Anda.
- Pastikan bahwa peran Anda memiliki izin yang ditunjukkan dalam contoh berikut.

Example kebijakan izin:

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

Ganti amzn-s3-demo-bucket dengan nama bucket Amazon S3 Anda.

- Jika bucket Amazon S3 dienkripsi menggunakan kunci KMS yang dikelola pelanggan, kunci KMS harus memiliki kebijakan akses dengan peran dan operasi IAM. kms:Decrypt kms:GenerateDataKey

- d. Untuk mengatur hot tier, lihat Langkah 5 di [Konfigurasi pengaturan penyimpanan untuk tingkat hangat \(konsol\)](#).
- e. (Opsional) Untuk AWS IoT Analytics integrasi, lakukan hal berikut.

 Note

Pemberitahuan akhir dukungan: Pada 15 Desember 2025, AWS akan mengakhiri dukungan untuk AWS IoT Analytics. Setelah 15 Desember 2025, Anda tidak akan lagi dapat mengakses AWS IoT Analytics konsol atau AWS IoT Analytics sumber daya. Untuk informasi selengkapnya, lihat [AWS IoT Analytics akhir dukungan](#).

- i. Jika Anda AWS IoT Analytics ingin menggunakan kueri data Anda, pilih Penyimpanan AWS IoT Analytics data yang diaktifkan.
- ii. AWS IoT SiteWise menghasilkan nama untuk penyimpanan data Anda atau Anda dapat memasukkan nama yang berbeda.

AWS IoT SiteWise secara otomatis membuat penyimpanan data AWS IoT Analytics untuk menyimpan data Anda. Untuk menanyakan data, Anda dapat menggunakan AWS IoT Analytics untuk membuat kumpulan data. Untuk informasi selengkapnya, lihat [Bekerja dengan AWS IoT SiteWise data](#) di Panduan AWS IoT Analytics Pengguna.

- f. Pilih Simpan.

Di bagian AWS IoT SiteWise penyimpanan, penyimpanan tingkat dingin dapat menjadi salah satu dari nilai berikut:

- Diaktifkan - AWS IoT SiteWise mereplikasi data Anda ke bucket Amazon S3 yang ditentukan.
- Mengaktifkan — AWS IoT SiteWise memproses permintaan Anda untuk mengaktifkan penyimpanan tingkat dingin. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya.
- Enable_Failed — AWS IoT SiteWise tidak dapat memproses permintaan Anda untuk mengaktifkan penyimpanan tingkat dingin. Jika Anda mengaktifkan AWS IoT SiteWise untuk mengirim log ke Amazon CloudWatch Logs, Anda dapat menggunakan log ini untuk memecahkan masalah. Untuk informasi selengkapnya, lihat [Monitor dengan Amazon CloudWatch Logs](#).
- Dinonaktifkan - Penyimpanan tingkat dingin dinonaktifkan.

Konfigurasi pengaturan penyimpanan untuk tingkat dingin (AWS CLI)

Prosedur berikut menunjukkan cara mengkonfigurasi pengaturan penyimpanan untuk mereplikasi data ke tingkat dingin menggunakan AWS CLI.

Untuk mengkonfigurasi pengaturan penyimpanan menggunakan AWS CLI

1. Untuk mengeksport data ke bucket Amazon S3 di akun Anda, jalankan perintah berikut untuk mengkonfigurasi pengaturan penyimpanan. Ganti *file-name* dengan nama file yang berisi konfigurasi AWS IoT SiteWise penyimpanan.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise konfigurasi penyimpanan

- Ganti *amzn-s3-demo-bucket* dengan nama bucket Amazon S3 Anda.
- Ganti *prefix* dengan awalan Amazon S3 Anda.
- Ganti *aws-account-id* dengan ID AWS akun Anda.
- Ganti *role-name* dengan nama peran akses Amazon S3 yang memungkinkan AWS IoT SiteWise untuk mengirim data ke Amazon S3.
- Ganti *retention-in-days* dengan bilangan bulat lebih besar dari atau sama dengan 30 hari.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "retentionPeriod": {
    "numberOfDays": retention-in-days,
    "unlimited": false
  }
}
```


 Note

- Anda harus menggunakan nama bucket Amazon S3 yang sama dalam konfigurasi AWS IoT SiteWise penyimpanan dan kebijakan IAM.
- Pastikan bahwa peran Anda memiliki izin yang ditunjukkan dalam contoh berikut.

Example kebijakan izin:

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

Ganti amzn-s3-demo-bucket dengan nama bucket Amazon S3 Anda.

- Jika bucket Amazon S3 dienkripsi menggunakan kunci KMS yang dikelola pelanggan, kunci KMS harus memiliki kebijakan akses dengan peran dan operasi IAM.
kms:Decrypt kms:GenerateDataKey

Example response

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "retentionPeriod": {
```

```
    "numberOfDays": 100,  
    "unlimited": false  
  },  
  "configurationStatus": {  
    "state": "UPDATE_IN_PROGRESS"  
  }  
}
```

Note

Diperlukan beberapa menit AWS IoT SiteWise untuk memperbarui konfigurasi penyimpanan.

2. Untuk mengambil informasi konfigurasi penyimpanan, jalankan perintah berikut.

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

Example response

```
{  
  "storageType": "MULTI_LAYER_STORAGE",  
  "multiLayerStorage": {  
    "customerManagedS3Storage": {  
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-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"  
}
```

3. Untuk berhenti mengeksport data ke bucket Amazon S3, jalankan perintah berikut untuk mengonfigurasi pengaturan penyimpanan.

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

Note

Secara default, data Anda hanya disimpan di tingkat panas AWS IoT SiteWise.

Example response

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

4. Untuk mengambil informasi konfigurasi penyimpanan, jalankan perintah berikut.

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

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

(Optional) Buat penyimpanan AWS IoT Analytics data (AWS CLI)**Note**

Pemberitahuan akhir dukungan: Pada 15 Desember 2025, AWS akan mengakhiri dukungan untuk AWS IoT Analytics. Setelah 15 Desember 2025, Anda tidak akan lagi dapat mengakses AWS IoT Analytics konsol atau AWS IoT Analytics sumber daya. Untuk informasi selengkapnya, lihat [AWS IoT Analytics akhir dukungan](#).

Penyimpanan AWS IoT Analytics data adalah repositori yang dapat diskalakan dan dapat dikueri yang menerima dan menyimpan data. Anda dapat menggunakan AWS IoT SiteWise konsol atau AWS IoT Analytics APIs membuat penyimpanan AWS IoT Analytics data untuk menyimpan AWS IoT SiteWise data Anda. Untuk menanyakan data, Anda membuat kumpulan data dengan menggunakan AWS IoT Analytics Untuk informasi selengkapnya, lihat [Bekerja dengan AWS IoT SiteWise data](#) di Panduan AWS IoT Analytics Pengguna.

Langkah-langkah berikut digunakan AWS CLI untuk membuat penyimpanan data di AWS IoT Analytics.

Untuk membuat penyimpanan data, jalankan perintah berikut. Ganti *file-name* dengan nama file yang berisi konfigurasi penyimpanan data.

```
aws iotanalytics create-datastore --cli-input-json file://file-name.json
```

 Note

- Anda harus menentukan nama bucket Amazon S3 yang ada. Jika Anda tidak memiliki ember Amazon S3, buat ember terlebih dahulu. Untuk informasi selengkapnya, lihat [Membuat bucket S3 pertama Anda](#) di Panduan Pengguna Amazon S3.
- Anda harus menggunakan nama bucket Amazon S3 yang sama dalam konfigurasi AWS IoT SiteWise penyimpanan, kebijakan IAM, dan konfigurasi penyimpanan AWS IoT Analytics data.

Example AWS IoT Analytics konfigurasi penyimpanan data

Ganti *data-store-name* dan *amzn-s3-demo-bucket* dengan nama penyimpanan AWS IoT Analytics data dan nama bucket Amazon S3 Anda.

```
{
  "datastoreName": "data-store-name",
  "datastoreStorage": {
    "iotSiteWiseMultiLayerStorage": {
      "customerManagedS3Storage": {
        "bucket": "amzn-s3-demo-bucket"
      }
    }
  },
}
```

```
"retentionPeriod": {  
  "numberOfDays": 90  
}  
}
```

Example response

```
{  
  "datastoreName": "datastore_IoTSiteWise_demo",  
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/  
datastore_IoTSiteWise_demo",  
  "retentionPeriod": {  
    "numberOfDays": 90,  
    "unlimited": false  
  }  
}
```

Memecahkan masalah pengaturan penyimpanan untuk AWS IoT SiteWise

Gunakan informasi berikut untuk memecahkan masalah dan menyelesaikan masalah dengan konfigurasi penyimpanan.

Masalah

- [Kesalahan: Bucket tidak ada](#)
- [Kesalahan: Akses ditolak ke jalur Amazon S3](#)
- [Kesalahan: Peran ARN tidak dapat diasumsikan](#)
- [Kesalahan: Gagal mengakses bucket Amazon S3 lintas wilayah](#)

Kesalahan: Bucket tidak ada

Solusi: AWS IoT SiteWise tidak dapat menemukan ember Amazon S3 Anda. Pastikan Anda memasukkan nama bucket Amazon S3 yang ada di Wilayah saat ini.

Kesalahan: Akses ditolak ke jalur Amazon S3

Solusi: AWS IoT SiteWise tidak dapat mengakses bucket Amazon S3 Anda. Lakukan hal-hal berikut:

- Pastikan Anda menggunakan bucket Amazon S3 yang sama dengan yang Anda tentukan dalam kebijakan IAM.
- Pastikan bahwa peran Anda memiliki izin yang ditunjukkan dalam contoh berikut.

Example kebijakan izin

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

Ganti amzn-s3-demo-bucket dengan nama bucket Amazon S3 Anda.

Kesalahan: Peran ARN tidak dapat diasumsikan

Solusi: AWS IoT SiteWise tidak dapat mengambil peran IAM atas nama Anda. Pastikan peran Anda mempercayai layanan berikut: `iotsitewise.amazonaws.com`. Untuk informasi selengkapnya, lihat [Saya tidak dapat mengambil peran](#), lihat Panduan Pengguna IAM.

Kesalahan: Gagal mengakses bucket Amazon S3 lintas wilayah

Solusi: Bucket Amazon S3 yang Anda tentukan berada di Wilayah yang berbeda AWS . Pastikan bucket dan AWS IoT SiteWise aset Amazon S3 Anda berada di Wilayah yang sama.

Jalur file dan skema data yang disimpan di tingkat dingin

AWS IoT SiteWise menyimpan data Anda di tingkat dingin dengan mereplikasi deret waktu, termasuk pengukuran, metrik, transformasi dan agregat, serta definisi model aset dan aset. Berikut ini menjelaskan jalur file dan skema data yang dikirim ke tingkat dingin.

Topik

- [Data peralatan \(pengukuran\)](#)
- [Metrik, transformasi, dan agregat](#)
- [Metadata aset](#)
- [Metadata hierarki aset](#)
- [File indeks data penyimpanan](#)

Data peralatan (pengukuran)

AWS IoT SiteWise mengeksport data peralatan (pengukuran) ke tingkat dingin setiap enam jam sekali. Data mentah disimpan dalam tingkat dingin dalam format [Apache AVRO](#) (.avro).

Jalur berkas

AWS IoT SiteWise menyimpan data peralatan (pengukuran) di tingkat dingin menggunakan templat berikut.

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/  
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Setiap jalur file ke data mentah di Amazon S3 berisi komponen berikut.

Komponen jalur	Deskripsi
keyPrefix	Awalan Amazon S3 yang Anda tentukan dalam konfigurasi penyimpanan. AWS IoT SiteWise Amazon S3 menggunakan awalan sebagai nama folder di bucket.

Komponen jalur	Deskripsi
<code>raw</code>	Folder yang menyimpan data deret waktu dari peralatan (pengukuran). <code>raw</code> Folder disimpan di folder awalan.
<code>seriesBucket</code>	Angka heksadesimal antara 00 dan ff. Angka ini berasal dari <code>timeSeriesId</code> . Partisi ini digunakan untuk meningkatkan throughput saat AWS IoT SiteWise menulis ke tingkat dingin. Saat Anda menggunakan Amazon Athena untuk menjalankan kueri, Anda dapat menggunakan partisi untuk partisi butir halus guna meningkatkan kinerja kueri. <code>seriesBucket</code> dan <code>timeSeriesBucket</code> dalam metadata aset adalah angka yang sama.
<code>startYear</code>	Tahun waktu mulai eksklusif yang terkait dengan data deret waktu.
<code>startMonth</code>	Bulan waktu mulai eksklusif yang terkait dengan data deret waktu.
<code>startDay</code>	Hari bulan dari waktu mulai eksklusif yang terkait dengan data deret waktu.

Komponen jalur	Deskripsi
<code>fileName</code>	Nama file menggunakan karakter garis bawah (_) sebagai pembatas untuk memisahkan yang berikut: <code>rawAwalan</code>. <code>timeSeriesId</code> Nilainya. - Stempel waktu zaman dari waktu mulai eksklusif yang terkait dengan data deret waktu. - Kualitas data. Nilai-nilai yang valid: GOOD, BAD, dan UNCERTAIN . Untuk informasi selengkapnya, lihat AssetPropertyValue di dalam Referensi API AWS IoT SiteWise . File disimpan dalam <code>.avro</code> format dengan menggunakan kompresi Snappy.

Example jalur file ke data mentah di tingkat dingin

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_GOOD.avro
```

Bidang

Skema data mentah yang diekspor ke tingkat dingin berisi bidang-bidang berikut.

AWS IoT SiteWise menyarankan pelanggan untuk menerapkan dukungan untuk evolusi skema pada sistem yang membaca data mentah dari tingkat dingin, karena mungkin ada bidang tambahan yang diperkenalkan di masa depan.

Data nol direpresentasikan sebagai semua bidang nilai menjadi nol. Namun, pelanggan akan tetap menerima tipe data yang benar saat melakukan query dengan AWS IoT SiteWise APIs.

Nama bidang	Jenis yang didukung	Jenis default	Deskripsi
<code>seriesId</code>	<code>string</code>	N/A	ID yang mengidentifikasi data deret waktu dari peralatan (pengukuran). Anda dapat menggunakan bidang ini untuk menggabungkan data mentah dan metadata aset dalam kueri.
<code>timeInSeconds</code>	<code>long</code>	N/A	Tanggal stempel waktu, dalam hitungan detik, dalam format epoch Unix. Data nanodetik pecahan disediakan oleh <code>offsetInNanos</code>
<code>offsetInNanos</code>	<code>long</code>	N/A	Nanodetik offset dari <code>timeInSeconds</code>
<code>quality</code>	<code>string</code>	N/A	Kualitas nilai deret waktu.
<code>doubleValue</code>	<code>double</code> atau <code>null</code>	<code>null</code>	Data deret waktu tipe ganda (nomor floating point).
<code>stringValue</code>	<code>string</code> atau <code>null</code>	<code>null</code>	Data deret waktu dari tipe string (urutan karakter).
<code>integerValue</code>	<code>int</code> atau <code>null</code>	<code>null</code>	Data deret waktu tipe integer (bilangan bulat).

Nama bidang	Jenis yang didukung	Jenis default	Deskripsi
booleanValue	boolean atau null	null	Data deret waktu tipe Boolean (benar atau salah).
jsonValue	string atau null	null	Data deret waktu tipe JSON (tipe data kompleks yang disimpan sebagai string).
recordVersion	long atau null	null	Nomor versi untuk catatan. Anda dapat menggunakan nomor versi untuk memilih catatan terbaru. Catatan yang lebih baru memiliki nomor versi yang lebih besar.

Example data mentah di tingkat dingin

```
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675887,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {"double": 0.75},
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": null
},
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675889,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {"double": 0.69},
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": null
},
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675890,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {"double": 0.66},
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": null
},
{
  "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,
  "recordVersion": null
}
```

```
{ "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
```

Metrik, transformasi, dan agregat

AWS IoT SiteWise mengeksport metrik, transformasi, dan agregat ke tingkat dingin setiap enam jam sekali. Metrik, transformasi, dan agregat disimpan dalam tingkat dingin dalam format [Apache AVRO](#) (). `.avro`

Jalur berkas

AWS IoT SiteWise menyimpan metrik, transformasi, dan agregat di tingkat dingin menggunakan templat berikut.

```
{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Setiap jalur file ke metrik, transformasi, dan agregat di Amazon S3 berisi komponen berikut.

Komponen jalur	Deskripsi
keyPrefix	Awalan Amazon S3 yang Anda tentukan dalam konfigurasi penyimpanan. AWS IoT SiteWise Amazon S3 menggunakan awalan sebagai nama folder di bucket.
agg	Folder yang menyimpan data deret waktu dari metrik. aggFolder disimpan di folder awalan.
seriesBucket	Angka heksadesimal antara 00 dan ff. Angka ini berasal dari <code>timeSeriesId</code> . Partisi ini digunakan untuk meningkatkan throughput saat AWS IoT SiteWise menulis ke tingkat dingin. Saat Anda menggunakan Amazon Athena untuk menjalankan kueri, Anda dapat menggunakan partisi untuk partisi butir halus guna meningkatkan kinerja kueri.

Komponen jalur	Deskripsi
	<code>seriesBucket</code> dan <code>timeSeriesBucket</code> dalam metadata aset adalah angka yang sama.
<code>startYear</code>	Tahun waktu mulai eksklusif yang terkait dengan data deret waktu.
<code>startMonth</code>	Bulan waktu mulai eksklusif yang terkait dengan data deret waktu.
<code>startDay</code>	Hari bulan dari waktu mulai eksklusif yang terkait dengan data deret waktu.
<code>fileName</code>	Nama file menggunakan karakter garis bawah (<code>_</code>) sebagai pembatas untuk memisahkan yang berikut: <code>rawAwalan</code>. <code>timeSeriesId</code> Nilainya. - Stempel waktu zaman dari waktu mulai eksklusif yang terkait dengan data deret waktu. - Kualitas data. Nilai-nilai yang valid: <code>GOOD</code>, <code>BAD</code>, dan <code>UNCERTAIN</code> . Untuk informasi selengkapnya, lihat AssetPropertyValue di dalam Referensi API AWS IoT SiteWise . File disimpan dalam <code>.avro</code> format dengan menggunakan kompresi Snappy.

Example jalur file ke metrik di tingkat dingin

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/agg_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1609577700_GOOD.avro
```

Bidang

Skema metrik, transformasi, dan agregat yang diekspor ke tingkat dingin berisi bidang berikut.

Nama bidang	Jenis yang didukung	Jenis default	Deskripsi
<code>seriesId</code>	<code>string</code>	N/A	ID yang mengidentifikasi data deret waktu dari peralatan, metrik, atau transformasi. Anda dapat menggunakan bidang ini untuk menggabungkan data mentah dan metadata aset dalam kueri.
<code>timeInSeconds</code>	<code>long</code>	N/A	Tanggal stempel waktu, dalam hitungan detik, dalam format epoch Unix. Data nanodetik pecahan disediakan oleh <code>offsetInNanos</code>
<code>offsetInNanos</code>	<code>long</code>	N/A	Nanodetik offset dari <code>timeInSeconds</code>
<code>quality</code>	<code>string</code>	N/A	Kualitas yang digunakan untuk memfilter data aset.
<code>resolution</code>	<code>string</code>	N/A	Interval waktu untuk mengumpulkan data.
<code>count</code>	<code>double</code> atau <code>null</code>	<code>null</code>	Jumlah total titik data untuk variabel yang

Nama bidang	Jenis yang didukung	Jenis default	Deskripsi
			diberikan selama interval waktu saat ini.
average	double atau null	null	Rata-rata nilai variabel yang diberikan selama interval waktu saat ini.
min	double atau null	null	Minimum nilai variabel yang diberikan selama interval waktu saat ini.
max	boolean atau null	null	Maksimum nilai variabel yang diberikan selama interval waktu saat ini.
sum	string atau null	null	Jumlah nilai variabel yang diberikan selama interval waktu saat ini.
recordVersion	long atau null	null	Nomor versi untuk catatan. Anda dapat menggunakan nomor versi untuk memilih catatan terbaru. Catatan yang lebih baru memiliki nomor versi yang lebih besar.

Example Data metrik di tingkat dingin

```
{
  "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}
```

Metadata aset

Saat Anda mengaktifkan AWS IoT SiteWise untuk mengeksport data ke tingkat dingin untuk pertama kalinya, metadata aset diekspor ke tingkat dingin. Setelah konfigurasi awal, AWS IoT SiteWise ekspor metadata aset ke tingkat hanya jika Anda mengubah definisi model aset atau definisi aset. Metadata aset disimpan dalam tingkat dingin dalam format JSON () yang dibatasi baris baru. .ndjson

Jalur berkas

AWS IoT SiteWise menyimpan metadata aset di tingkat dingin menggunakan templat berikut.

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

Setiap jalur file ke metadata aset di tingkat dingin berisi komponen berikut.

Komponen jalur	Deskripsi
<code>keyPrefix</code>	Awalan Amazon S3 yang Anda tentukan dalam konfigurasi penyimpanan AWS IoT SiteWise. Amazon S3 menggunakan awalan sebagai nama folder di bucket.
<code>asset_metadata</code>	Folder yang menyimpan metadata aset. <code>asset_metadata</code> Folder disimpan di folder awalan.
<code>fileName</code>	Nama file menggunakan karakter garis bawah (_) sebagai pembatas untuk memisahkan yang berikut: <code>assetAwalan</code>. <code>assetIdNilainya</code>. File disimpan dalam <code>.ndjson</code> format.

Example jalur file ke metadata aset di tingkat yang lebih dingin

`keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson`

Bidang

Skema metadata aset yang diekspor ke tingkat dingin berisi bidang-bidang berikut.

Nama bidang	Deskripsi
<code>assetId</code>	ID aset.
<code>assetName</code>	Nama aset.
<code>assetExternalId</code>	ID eksternal aset.
<code>assetModelId</code>	ID model aset yang digunakan untuk membuat aset ini.

Nama bidang	Deskripsi
<code>assetModelName</code>	Nama model aset.
<code>assetModelExternalId</code>	ID eksternal dari model aset.
<code>assetPropertyId</code>	ID dari properti aset.
<code>assetPropertyName</code>	Nama properti aset.
<code>assetPropertyExternalId</code>	ID eksternal dari properti aset.
<code>assetPropertyDataType</code>	Tipe data properti aset.
<code>assetPropertyUnit</code>	Unit properti aset (misalnya, Newtons dan RPM).
<code>assetPropertyAlias</code>	Alias yang mengidentifikasi properti aset, seperti jalur aliran data server OPC UA (misalnya,). <code>/company/windfarm/3/turbine/7/temperature</code>
<code>timeSeriesId</code>	ID yang mengidentifikasi data deret waktu dari peralatan, metrik, atau transformasi. Anda dapat menggunakan bidang ini untuk menggabungkan data mentah dan metadata aset dalam kueri.

Nama bidang	Deskripsi
<code>timeSeriesBucket</code>	Angka heksadesimal antara 00 dan ff. Angka ini berasal dari <code>timeSeriesId</code> . Partisi ini digunakan untuk meningkatkan throughput saat AWS IoT SiteWise menulis ke tingkat dingin. Saat Anda menggunakan Amazon Athena untuk menjalankan kueri, Anda dapat menggunakan partisi untuk partisi butir halus guna meningkatkan kinerja kueri. <code>timeSeriesBucket</code> dan <code>seriesBucket</code> di jalur file ke data mentah adalah nomor yang sama.
<code>assetCompositeModelId</code>	ID dari model komposit.
<code>assetCompositeModelExternalId</code>	ID eksternal dari model komposit.
<code>assetCompositeModelDescription</code>	Deskripsi dari model komposit.
<code>assetCompositeModelName</code>	Nama dari model komposit.
<code>assetCompositeModelType</code>	Jenis dari model komposit. Untuk model komposit alarm, jenis ini berupa <code>AWS/ALARM</code> .
<code>assetCreationDate</code>	Tanggal aset dibuat, dalam waktu zaman Unix.
<code>assetLastUpdateDate</code>	Tanggal aset terakhir diperbarui, dalam waktu zaman Unix.
<code>assetStatusErrorCode</code>	Kode kesalahan.
<code>assetStatusErrorMessage</code>	Pesan kesalahan.
<code>assetStatusState</code>	Status aset saat ini.

Example metadata aset di tingkat dingin

```
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d",
  "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",
  "assetPropertyData": {
    "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a",
    "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a",
    "timeSeriesBucket": "f6",
    "assetArn": null,
    "assetCompositeModelDescription": null
  },
  "assetPropertyUnit": "C",
  "assetPropertyDataType": "DOUBLE",
  "assetPropertyAlias": null
},
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d",
  "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,
  "assetPropertyName": "Pressure",
  "assetPropertyData": {
    "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4",
    "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4",
    "timeSeriesBucket": "1e",
    "assetArn": null,
    "assetCompositeModelDescription": null
  },
  "assetPropertyUnit": "Pa",
  "assetPropertyDataType": "DOUBLE",
  "assetPropertyAlias": null
},
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d",
  "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",
  "assetPropertyData": {
    "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_8cf1162f-dead-4fbe-b468-c8e24cde9f50",
    "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_8cf1162f-dead-4fbe-b468-c8e24cde9f50",
    "timeSeriesBucket": "d7",
    "assetArn": null,
    "assetCompositeModelDescription": null,
    "assetPropertyUnit": "C"
  },
  "assetPropertyUnit": "C",
  "assetPropertyDataType": "DOUBLE",
  "assetPropertyAlias": null
},
{
  "assetId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab",
  "assetExternalId": null,
  "assetName": "BatchAsset",
  "assetModelId": "ebc75e75e827",
  "assetModelExternalId": null,
  "assetModelName": "FlashTestAssetModelDouble",
  "assetPropertyId": "ab401a9176ed",
  "assetPropertyExternalId": null,
  "assetPropertyName": "measurementProperty",
  "assetPropertyData": {
    "assetId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab_ab401a9176ed",
    "timeSeriesId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab_ab401a9176ed",
    "timeSeriesBucket": "af",
    "assetArn": null,
    "assetCompositeModelDescription": null,
    "assetPropertyUnit": "C"
  },
  "assetPropertyUnit": "C",
  "assetPropertyDataType": "DOUBLE",
  "assetPropertyAlias": null
}
```

Metadata hierarki aset

Saat Anda mengaktifkan AWS IoT SiteWise untuk menyimpan data dalam tingkat dingin untuk pertama kalinya, metadata hierarki aset diekspor ke tingkat dingin. Setelah konfigurasi awal, AWS IoT SiteWise mengeksport metadata hierarki aset ke tingkat dingin hanya jika Anda membuat perubahan

pada model aset atau definisi aset. Metadata hierarki aset disimpan dalam tingkat dingin dalam format JSON () yang dibatasi baris baru. .ndjson

Pengidentifikasi eksternal untuk hierarki, aset target, atau aset sumber diambil dengan memanggil API. [DescribeAsset](#)

Jalur berkas

AWS IoT SiteWise menyimpan metadata hierarki aset di tingkat dingin menggunakan templat berikut.

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

Setiap jalur file ke metadata hierarki aset di tingkat dingin berisi komponen berikut.

Komponen jalur	Deskripsi
keyPrefix	Awalan Amazon S3 yang Anda tentukan dalam konfigurasi penyimpanan. AWS IoT SiteWise Amazon S3 menggunakan awalan sebagai nama folder di bucket.
asset_hierarchy_metadata	Folder yang menyimpan metadata hierarki aset. asset_hierarchy_metadata Folder disimpan di folder awalan.
fileName	Nama file menggunakan karakter garis bawah (_) sebagai pembatas untuk memisahkan yang berikut: - parentAssetId Nilainya. - hierarchyId Nilainya. File disimpan dalam .ndjson format.

Example jalur file ke metadata hierarki aset di tingkat dingin

```
keyPrefix/asset_hierarchy_metadata/35901915-d476-4dca-8637-d9ed4df939ed_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson
```

Bidang

Skema metadata hierarki aset yang diekspor ke tingkat dingin berisi bidang berikut.

Nama bidang	Deskripsi
sourceAssetId	ID aset sumber dalam hubungan aset ini.
targetAssetId	ID aset target dalam hubungan aset ini.
hierarchyId	ID hierarki.
associationType	Jenis asosiasi dari hubungan aset ini. Nilainya harusCHILD. Aset target adalah aset anak dari aset sumber.

Example metadata hierarki aset di tingkat dingin

```
{"sourceAssetId":"80388e72-2284-44fb-9c89-bfbaf0dfedd2","targetAssetId":"2b866c25-0c74-4750-bdf5-b73683c8a2a2","hierarchyId":"bbled9f59-0412-4585-a61d-6044db526aee","associationType":"CHILD"}
{"sourceAssetId":"80388e72-2284-44fb-9c89-bfbaf0dfedd2","targetAssetId":"6b51246e-984d-460d-bc0b-470ea47d1e31","hierarchyId":"bbled9f59-0412-4585-a61d-6044db526aee","associationType":"CHILD"}
```

Untuk melihat data Anda di tingkat dingin

1. Arahkan ke [konsol Amazon S3](#).
2. Di panel navigasi, pilih Bucket, lalu pilih bucket Amazon S3 Anda.
3. Arahkan ke folder yang berisi data mentah, metadata aset, atau metadata hierarki aset.
4. Pilih file, dan kemudian dari Tindakan, pilih Unduh.

File indeks data penyimpanan

AWS IoT SiteWise menggunakan file-file ini untuk mengoptimalkan kinerja kueri data. Mereka muncul di ember Amazon S3 Anda, tetapi Anda tidak perlu menggunakannya.

Jalur berkas

AWS IoT SiteWise menyimpan file indeks data di tingkat dingin menggunakan template berikut.

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/  
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Example jalur file ke file indeks penyimpanan data

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-  
d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/  
index_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-  
bc6f-1b490154b07a_1643846400_G00D
```

Contoh kode untuk AWS IoT SiteWise menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan AWS IoT SiteWise kit pengembangan AWS perangkat lunak (SDK).

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Memulai

Halo AWS IoT SiteWise

Contoh kode berikut menunjukkan cara untuk mulai menggunakan AWS IoT SiteWise.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
public class HelloSitewise {  
    private static final Logger logger =  
        LoggerFactory.getLogger(HelloSitewise.class);  
    public static void main(String[] args) {  
        fetchAssetModels();  
    }  
}
```



```

/**
 * Fetches asset models using the provided {@link IoTSiteWiseAsyncClient}.
 */
public static void fetchAssetModels() {
    IoTSiteWiseAsyncClient siteWiseAsyncClient =
IoTSiteWiseAsyncClient.create();
    ListAssetModelsRequest assetModelsRequest =
ListAssetModelsRequest.builder()
        .assetModelTypes(AssetModelType.ASSET_MODEL)
        .build();

    // Asynchronous paginator - process paginated results.
    ListAssetModelsPublisher listModelsPaginator =
siteWiseAsyncClient.listAssetModelsPaginator(assetModelsRequest);
    CompletableFuture<Void> future = listModelsPaginator.subscribe(response -
> {
        response.assetModelSummaries().forEach(assetSummary ->
            logger.info("Asset Model Name: {} ", assetSummary.name())
        );
    });

    // Wait for the asynchronous operation to complete
    future.join();
}
}

```

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    paginateListAssetModels,

```

```

    IoTSiteWiseClient,
  } from "@aws-sdk/client-iotsitewise";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new IoTSiteWiseClient();
  const listAssetModelsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListAssetModels({ client }, { maxResults: 5 });
    for await (const page of paginator) {
      listAssetModelsPaginated.push(...page.assetModelSummaries);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }
  for (const { name, creationDate } of listAssetModelsPaginated) {
    console.log(`${name} - ${creationDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}

```

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import boto3

def hello_iot_sitewise(iot_sitewise_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS IoT SiteWise
    client and list the asset models in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param iot_sitewise_client: A Boto3 AWS IoT SiteWise Client object. This
    object wraps
                                the low-level AWS IoT SiteWise service API.
    """
    print("Hello, AWS IoT SiteWise! Let's list some of your asset models:\n")
    paginator = iot_sitewise_client.get_paginator("list_asset_models")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 10})

    asset_model_names: [str] = []
    for page in page_iterator:
        for asset_model in page["assetModelSummaries"]:
            asset_model_names.append(asset_model["name"])

    print(f"{len(asset_model_names)} asset model(s) retrieved.")
    for asset_model_name in asset_model_names:
        print(f"\t{asset_model_name}")

if __name__ == "__main__":
    hello_iot_sitewise(boto3.client("iotsitewise"))

```

- Untuk detail API, lihat [ListAssetModels](#) di AWS SDK for Python (Boto3) Referensi API.

Contoh kode

- [Contoh dasar untuk AWS IoT SiteWise menggunakan AWS SDKs](#)
 - [Halo AWS IoT SiteWise](#)
 - [Pelajari dasar-dasar AWS IoT SiteWise dengan AWS SDK](#)
 - [Tindakan untuk AWS IoT SiteWise menggunakan AWS SDKs](#)
 - [Gunakan BatchPutAssetPropertyValue dengan AWS SDK atau CLI](#)

- [Gunakan CreateAsset dengan AWS SDK atau CLI](#)
- [Gunakan CreateAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan CreateGateway dengan AWS SDK atau CLI](#)
- [Gunakan CreatePortal dengan AWS SDK atau CLI](#)
- [Gunakan DeleteAsset dengan AWS SDK atau CLI](#)
- [Gunakan DeleteAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan DeleteGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeletePortal dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan DescribeGateway dengan AWS SDK atau CLI](#)
- [Gunakan DescribePortal dengan AWS SDK atau CLI](#)
- [Gunakan GetAssetPropertyValue dengan AWS SDK atau CLI](#)
- [Gunakan ListAssetModels dengan AWS SDK atau CLI](#)

Contoh dasar untuk AWS IoT SiteWise menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar AWS IoT SiteWise dengan AWS SDKs.

Contoh

- [Halo AWS IoT SiteWise](#)
- [Pelajari dasar-dasar AWS IoT SiteWise dengan AWS SDK](#)
- [Tindakan untuk AWS IoT SiteWise menggunakan AWS SDKs](#)
 - [Gunakan BatchPutAssetPropertyValue dengan AWS SDK atau CLI](#)
 - [Gunakan CreateAsset dengan AWS SDK atau CLI](#)
 - [Gunakan CreateAssetModel dengan AWS SDK atau CLI](#)
 - [Gunakan CreateGateway dengan AWS SDK atau CLI](#)
 - [Gunakan CreatePortal dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteAsset dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteAssetModel dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteGateway dengan AWS SDK atau CLI](#)

- [Gunakan DeletePortal dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan DescribeGateway dengan AWS SDK atau CLI](#)
- [Gunakan DescribePortal dengan AWS SDK atau CLI](#)
- [Gunakan GetAssetPropertyValue dengan AWS SDK atau CLI](#)
- [Gunakan ListAssetModels dengan AWS SDK atau CLI](#)

Halo AWS IoT SiteWise

Contoh kode berikut menunjukkan cara untuk mulai menggunakan AWS IoT SiteWise.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
public class HelloSitewise {
    private static final Logger logger =
        LoggerFactory.getLogger(HelloSitewise.class);
    public static void main(String[] args) {
        fetchAssetModels();
    }

    /**
     * Fetches asset models using the provided {@link IoTSiteWiseAsyncClient}.
     */
    public static void fetchAssetModels() {
        IoTSiteWiseAsyncClient siteWiseAsyncClient =
            IoTSiteWiseAsyncClient.create();
        ListAssetModelsRequest assetModelsRequest =
            ListAssetModelsRequest.builder()
                .assetModelTypes(AssetModelType.ASSET_MODEL)
                .build();
```

```

        // Asynchronous paginator - process paginated results.
        ListAssetModelsPublisher listModelsPaginator =
siteWiseAsyncClient.listAssetModelsPaginator(assetModelsRequest);
        CompletableFuture<Void> future = listModelsPaginator.subscribe(response -
> {
            response.assetModelSummaries().forEach(assetSummary ->
                logger.info("Asset Model Name: {} ", assetSummary.name())
            );
        });

        // Wait for the asynchronous operation to complete
        future.join();
    }
}

```

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    paginateListAssetModels,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";

// Call ListDocuments and display the result.
export const main = async () => {
    const client = new IoTSiteWiseClient();
    const listAssetModelsPaginated = [];
    console.log(
        "Hello, AWS Systems Manager! Let's list some of your documents:\n",
    );
    try {
        // The paginate function is a wrapper around the base command.
    }
}

```

```

const paginator = paginateListAssetModels({ client }, { maxResults: 5 });
for await (const page of paginator) {
  listAssetModelsPaginated.push(...page.assetModelSummaries);
}
} catch (caught) {
  console.error(`There was a problem saying hello: ${caught.message}`);
  throw caught;
}
for (const { name, creationDate } of listAssetModelsPaginated) {
  console.log(`${name} - ${creationDate}`);
}
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}

```

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import boto3

def hello_iot_sitewise(iot_sitewise_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS IoT SiteWise
    client and list the asset models in your account.
    This example uses the default settings specified in your shared credentials
    and config files.
    """

```

```

:param iot_sitewise_client: A Boto3 AWS IoT SiteWise Client object. This
object wraps
                                the low-level AWS IoT SiteWise service API.

"""
print("Hello, AWS IoT SiteWise! Let's list some of your asset models:\n")
paginator = iot_sitewise_client.get_paginator("list_asset_models")
page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 10})

asset_model_names: [str] = []
for page in page_iterator:
    for asset_model in page["assetModelSummaries"]:
        asset_model_names.append(asset_model["name"])

print(f"{len(asset_model_names)} asset model(s) retrieved.")
for asset_model_name in asset_model_names:
    print(f"\t{asset_model_name}")

if __name__ == "__main__":
    hello_iot_sitewise(boto3.client("iotsitewise"))

```

- Untuk detail API, lihat [ListAssetModels](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Pelajari dasar-dasar AWS IoT SiteWise dengan AWS SDK

Contoh-contoh kode berikut menunjukkan cara:

- Buat Model AWS IoT SiteWise Aset.
- Buat AWS IoT SiteWise Aset.
- Mengambil nilai ID properti.
- Kirim data ke AWS IoT SiteWise Aset.
- Ambil nilai properti AWS IoT SiteWise Aset.
- Buat AWS IoT SiteWise Portal.
- Buat AWS IoT SiteWise Gateway.

- Jelaskan AWS IoT SiteWise Gateway.
- Hapus AWS IoT SiteWise Aset.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jalankan skenario interaktif yang menunjukkan AWS IoT SiteWise fitur.

```
public class SitewiseScenario {

    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");

    private static final Logger logger =
    LoggerFactory.getLogger(SitewiseScenario.class);
    static Scanner scanner = new Scanner(System.in);

    private static final String ROLES_STACK = "RoleSitewise";

    static SitewiseActions sitewiseActions = new SitewiseActions();

    public static void main(String[] args) throws Throwable {
        Scanner scanner = new Scanner(System.in);
        String contactEmail = "user@mydomain.com"; // Change email address.
        String assetModelName = "MyAssetModel1";
        String assetName = "MyAsset1" ;
        String portalName = "MyPortal1" ;
        String gatewayName = "MyGateway1" ;
        String myThing = "MyThing1" ;

        logger.info("""
            AWS IoT SiteWise is a fully managed software-as-a-service (SaaS)
            that
```

makes it easy to collect, store, organize, and monitor data from industrial equipment and processes.

It is designed to help industrial and manufacturing organizations collect data from their equipment and processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide range of industrial equipment and systems, including programmable logic controllers (PLCs), sensors, and other industrial devices. It can collect data from these devices and organize it into a unified data model, making it easier to analyze and gain insights from the data. AWS IoT SiteWise also provides tools for visualizing the data, setting up alarms and alerts, and generating reports.

Another key feature of AWS IoT SiteWise is its ability to scale to handle large volumes of data.

It can collect and store data from thousands of devices and process millions of data points per second, making it suitable for large-scale industrial operations. Additionally, AWS IoT SiteWise is designed to be secure and compliant, with features like role-based access controls, data encryption, and integration with other AWS services for additional security and compliance features.

Let's get started...

```
""");
```

```
waitForInputToContinue(scanner);
```

```
logger.info(DASHES);
```

```
try {
```

```
    runScenario(assetModelName, assetName, portalName, contactEmail,  
gatewayName, myThing);
```

```
    } catch (RuntimeException e) {
```

```
        logger.info(e.getMessage());
```

```
    }
```

```
}
```

```

    public static void runScenario(String assetModelName, String assetName,
String portalName, String contactEmail, String gatewayName, String myThing)
throws Throwable {
    logger.info("Use AWS CloudFormation to create an IAM role that is
required for this scenario.");
    CloudFormationHelper.deployCloudFormationStack(ROLES_STACK);
    Map<String, String> stackOutputs =
CloudFormationHelper.getStackOutputsAsync(ROLES_STACK).join();
    String iamRole = stackOutputs.get("SitewiseRoleArn");
    logger.info("The ARN of the IAM role is {}", iamRole);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("1. Create an AWS SiteWise Asset Model");
    logger.info("""
        An AWS IoT SiteWise Asset Model is a way to represent the physical
assets, such as equipment,
        processes, and systems, that exist in an industrial environment.
This model provides a structured and
        hierarchical representation of these assets, allowing users to
define the relationships and properties
        of each asset.

        This scenario creates two asset model properties: temperature and
humidity.
        """);
    waitForInputToContinue(scanner);
    String assetModelId = null;
    try {
        CreateAssetModelResponse response =
sitewiseActions.createAssetModelAsync(assetModelName).join();
        assetModelId = response.assetModelId();
        logger.info("Asset Model successfully created. Asset Model ID: {}. ",
assetModelId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceAlreadyExistsException) {
            try {
                assetModelId =
sitewiseActions.getAssetModelIdAsync(assetModelName).join();
                logger.info("The Asset Model {} already exists. The id of the
existing model is {}. Moving on...", assetModelName, assetModelId);
            } catch (CompletionException cex) {

```

```

        logger.error("Exception thrown acquiring the asset model id:
{}", cex.getCause().getCause(), cex);
        return;
    }
} else {
    logger.info("An unexpected error occurred: " +
cause.getMessage(), cause);
    return;
}
}
waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("2. Create an AWS IoT SiteWise Asset");
logger.info("""
    The IoT SiteWise model that we just created defines the structure
and metadata for your physical assets.
    Now we create an asset from the asset model.

    """);
logger.info("Let's wait 30 seconds for the asset to be ready.");
countdown(30);
waitForInputToContinue(scanner);
String assetId;
try {
    CreateAssetResponse response =
sitewiseActions.createAssetAsync(assetName, assetModelId).join();
    assetId = response.assetId();
    logger.info("Asset created with ID: {}", assetId);
} catch (CompletionException ce) {
    Throwable cause = ce.getCause();
    if (cause instanceof ResourceNotFoundException) {
        logger.info("The asset model id was not found: {}",
cause.getMessage(), cause);
    } else {
        logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
    }
    return;
}
waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);

```

```

        logger.info("3. Retrieve the property ID values");
        logger.info("""
            To send data to an asset, we need to get the property ID values. In
this scenario, we access the
            temperature and humidity property ID values.
            """);
        waitForInputToContinue(scanner);
        Map<String, String> propertyIds = null;
        try {
            propertyIds = sitewiseActions.getPropertyIds(assetModelId).join();
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof IoTSiteWiseException) {
                logger.error("IoTSiteWiseException occurred: {}",
cause.getMessage(), ce);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage(), ce);
            }
            return;
        }
        String humPropId = propertyIds.get("Humidity");
        logger.info("The Humidity property Id is {}", humPropId);
        String tempPropId = propertyIds.get("Temperature");
        logger.info("The Temperature property Id is {}", tempPropId);

        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("4. Send data to an AWS IoT SiteWise Asset");
        logger.info("""
            By sending data to an IoT SiteWise Asset, you can aggregate data
from
            multiple sources, normalize the data into a standard format, and
store it in a
            centralized location. This makes it easier to analyze and gain
insights from the data.

            In this example, we generate sample temperature and humidity data and
send it to the AWS IoT SiteWise asset.

            """);
        waitForInputToContinue(scanner);

```

```

        try {
            sitewiseActions.sendDataToSiteWiseAsync(assetId, tempPropId,
humPropId).join();
            logger.info("Data sent successfully.");
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException) {
                logger.error("The AWS resource was not found: {}",
cause.getMessage(), cause);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage(), cause);
            }
            return;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("5. Retrieve the value of the IoT SiteWise Asset property");
        logger.info("""
            IoT SiteWise is an AWS service that allows you to collect, process,
and analyze industrial data
            from connected equipment and sensors. One of the key benefits of
reading an IoT SiteWise property
            is the ability to gain valuable insights from your industrial data.

            """);
        waitForInputToContinue(scanner);
        try {
            Double assetVal = sitewiseActions.getAssetPropValueAsync(tempPropId,
assetId).join();
            logger.info("The property name is: {}", "Temperature");
            logger.info("The value of this property is: {}", assetVal);

            waitForInputToContinue(scanner);

            assetVal = sitewiseActions.getAssetPropValueAsync(humPropId,
assetId).join();
            logger.info("The property name is: {}", "Humidity");
            logger.info("The value of this property is: {}", assetVal);
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException) {

```

```

        logger.info("The AWS resource was not found: {}",
cause.getMessage(), cause);
    } else {
        logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
    }
    return;
}

waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("6. Create an IoT SiteWise Portal");
logger.info("""
    An IoT SiteWise Portal allows you to aggregate data from multiple
industrial sources,
    such as sensors, equipment, and control systems, into a centralized
platform.
    """);
waitForInputToContinue(scanner);
String portalId;
try {
    portalId = sitewiseActions.createPortalAsync(portalName, iamRole,
contactEmail).join();
    logger.info("Portal created successfully. Portal ID {}", portalId);
} catch (CompletionException ce) {
    Throwable cause = ce.getCause();
    if (cause instanceof IoTSiteWiseException siteWiseEx) {
        logger.error("IoT SiteWise error occurred: Error message: {},
Error code {}",
            siteWiseEx.getMessage(),
siteWiseEx.awsErrorDetails().errorCode(), siteWiseEx);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage());
    }
    return;
}

waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("7. Describe the Portal");
logger.info("""

```

In this step, we get a description of the portal and display the portal URL.

```

        """);
        waitForInputToContinue(scanner);
        try {
            String portalUrl =
sitewiseActions.describePortalAsync(portalId).join();
            logger.info("Portal URL: {}", portalUrl);
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException notFoundException) {
                logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                    notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage());
            }
            return;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("8. Create an IoT SiteWise Gateway");
        logger.info(
            ""

```

IoT SiteWise Gateway serves as the bridge between industrial equipment, sensors, and the cloud-based IoT SiteWise service. It is responsible for securely collecting, processing, and transmitting data from various industrial assets to the IoT SiteWise platform, enabling real-time monitoring, analysis, and optimization of industrial operations.

```

        """);
        waitForInputToContinue(scanner);
        String gatewayId = "";
        try {
            gatewayId = sitewiseActions.createGatewayAsync(gatewayName,
myThing).join();

```



```

        logger.info("Gateway creation completed successfully. id is {}",
gatewayId );
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof IoTSiteWiseException siteWiseEx) {
            logger.error("IoT SiteWise error occurred: Error message: {},
Error code {}",
                        siteWiseEx.getMessage(),
siteWiseEx.awsErrorDetails().errorCode(), siteWiseEx);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
        return;
    }
    logger.info(DASHES);
    logger.info(DASHES);

    logger.info("9. Describe the IoT SiteWise Gateway");
    waitForInputToContinue(scanner);
    try {
        sitewiseActions.describeGatewayAsync(gatewayId)
            .thenAccept(response -> {
                logger.info("Gateway Name: {}", response.gatewayName());
                logger.info("Gateway ARN: {}", response.gatewayArn());
                logger.info("Gateway Platform: {}",
response.gatewayPlatform());
                logger.info("Gateway Creation Date: {}",
response.creationDate());
            }).join();
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException) {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage(), cause);
        }
        return;
    }
    logger.info(DASHES);

```

```

logger.info(DASHES);
logger.info("10. Delete the AWS IoT SiteWise Assets");
logger.info(
    ""
    Before you can delete the Asset Model, you must delete the assets.

    "");
logger.info("Would you like to delete the IoT SiteWise Assets? (y/n)");
String delAns = scanner.nextLine().trim();
if (delAns.equalsIgnoreCase("y")) {
    logger.info("You selected to delete the SiteWise assets.");
    waitForInputToContinue(scanner);
    try {
        sitewiseActions.deletePortalAsync(portalId).join();
        logger.info("Portal {} was deleted successfully.", portalId);

    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }

    try {
        sitewiseActions.deleteGatewayAsync(gatewayId).join();
        logger.info("Gateway {} was deleted successfully.", gatewayId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {

```

```
        logger.error("An unexpected error occurred: {}",
cause.getMessage());
    }
}

    try {
        sitewiseActions.deleteAssetAsync(assetId).join();
        logger.info("Request to delete asset {} sent successfully",
assetId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }
    logger.info("Let's wait 1 minute for the asset to be deleted.");
    countdown(60);
    waitForInputToContinue(scanner);
    logger.info("Delete the AWS IoT SiteWise Asset Model");
    try {
        sitewiseActions.deleteAssetModelAsync(assetModelId).join();
        logger.info("Asset model deleted successfully.");
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }
    waitForInputToContinue(scanner);
```

```

    } else {
        logger.info("The resources will not be deleted.");
    }
    logger.info(DASHES);

    logger.info(DASHES);
    CloudFormationHelper.destroyCloudFormationStack(ROLES_STACK);
    logger.info("This concludes the AWS IoT SiteWise Scenario");
    logger.info(DASHES);
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        } else {
            logger.info("Invalid input. Please try again.");
        }
    }
}

public static void countdown(int totalSeconds) throws InterruptedException {
    for (int i = totalSeconds; i >= 0; i--) {
        int displayMinutes = i / 60;
        int displaySeconds = i % 60;
        System.out.printf("\r%02d:%02d", displayMinutes, displaySeconds);
        Thread.sleep(1000); // Wait for 1 second
    }
    System.out.println(); // Move to the next line after countdown
    logger.info("Countdown complete!");
}
}

```

Kelas pembungkus untuk metode AWS IoT SiteWise SDK.

```
public class SitewiseActions {
```

```

    private static final Logger logger =
LoggerFactory.getLogger(SitewiseActions.class);

    private static IoTSiteWiseAsyncClient ioTSiteWiseAsyncClient;

    private static IoTSiteWiseAsyncClient getAsyncClient() {
        if (ioTSiteWiseAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
                .apiCallTimeout(Duration.ofMinutes(2))
                .apiCallAttemptTimeout(Duration.ofSeconds(90))
                .retryStrategy(RetryMode.STANDARD)
                .build();

            ioTSiteWiseAsyncClient = IoTSiteWiseAsyncClient.builder()
                .httpClient(httpClient)
                .overrideConfiguration(overrideConfig)
                .build();
        }
        return ioTSiteWiseAsyncClient;
    }

    /**
     * Creates an asset model.
     *
     * @param name the name of the asset model to create.
     * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
     *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
     *         {@link CompletableFuture#get()}.
     * <p>
     *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it

```

```

    *         available to the calling code as a {@link CompletionException}. By
calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
        PropertyType humidity = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        PropertyType temperaturePropertyType = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
            .name("Temperature")
            .dataType(PropertyDataType.DOUBLE)
            .type(temperaturePropertyType)
            .build();

        AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
            .name("Humidity")
            .dataType(PropertyDataType.DOUBLE)
            .type(humidity)
            .build();

        CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
            .assetModelName(name)
            .assetModelDescription("This is my asset model")
            .assetModelProperties(temperatureProperty, humidityProperty)
            .build();

        return getAsyncClient().createAssetModel(createAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset model: {} ",
exception.getCause().getMessage());
                }
            });
    }
}

```

```

/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName    the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *
 * <p>
 *
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
    CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
        .assetModelId(assetModelId)
        .assetDescription("Created using the AWS SDK for Java")
        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
            }
        });
}

/**
 * Sends data to the SiteWise service.
 *
 * @param assetId        the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.

```

```

    * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
    *      calling code can attach callbacks, then handle the result or
exception by calling
    *      {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
    *      available to the calling code as a {@link CompletionException}. By
calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();

    BatchPutAssetPropertyValueRequest request =
BatchPutAssetPropertyValueRequest.builder()
        .entries(Arrays.asList(
            PutAssetPropertyValueEntry.builder()
                .entryId("entry-3")
                .assetId(assetId)
                .propertyId(tempPropertyId)
                .propertyValues(Arrays.asList(
                    AssetPropertyValue.builder()
                        .value(Variant.builder()
                            .doubleValue(sampleData.get("Temperature"))
                            .build())
                        .timestamp(time)
                        .build()
                ))
                .build(),
            PutAssetPropertyValueEntry.builder()
                .entryId("entry-4")
                .assetId(assetId)

```



```

        .propertyId(humidityPropId)
        .propertyValues(Arrays.asList(
            AssetPropertyValue.builder()
                .value(Variant.builder()
                    .doubleValue(sampleData.get("Humidity"))
                    .build())
                .timestamp(time)
                .build()
        ))
        .build()
    ))
    .build();

    return getAsyncClient().batchPutAssetPropertyValue(request)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An exception occurred: {}",
exception.getCause().getMessage());
            }
        });
}

/**
 * Fetches the value of an asset property.
 *
 * @param propId the ID of the asset property to fetch.
 * @param assetId the ID of the asset to fetch the property value for.
 * @return a {@link CompletableFuture} that represents a {@link Double}
result. The calling code can attach
 *      callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
    GetAssetPropertyValueRequest assetPropertyValueRequest =
GetAssetPropertyValueRequest.builder()

```

```

        .propertyId(propId)
        .assetId(assetId)
        .build();

    return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.propertyValue().value().doubleValue();
        });
}

/**
 * Retrieves the property IDs associated with a specific asset model.
 *
 * @param assetModelId the ID of the asset model that defines the properties.
 * @return a {@link CompletableFuture} that represents a {@link Map} result
that associates the property name to the
 *         propert ID. The calling code can attach callbacks, then handle the
result or exception by calling
 *         {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *
 * <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
    ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
    return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
        .handle((response, throwable) -> {
            if (response != null) {
                return response.assetModelPropertySummaries().stream()
                    .collect(Collectors
                        .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
            }
        });
}

```

```

        } else {
            logger.error("Error occurred while fetching property IDs:
{}.", throwable.getCause().getMessage());
            throw (CompletionException) throwable;
        }
    });
}

/**
 * Deletes an asset.
 *
 * @param assetId the ID of the asset to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
 *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
assetId) {
    DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
        .assetId(assetId)
        .build();

    return getAsyncClient().deleteAsset(deleteAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred deleting asset with id: {}",
assetId);
            }
        });
}

/**
 * Deletes an Asset Model with the specified ID.
 *
 * @param assetModelId the ID of the Asset Model to delete.

```

```

    * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
    *      can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
        DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
            .assetModelId(assetModelId)
            .build();

        return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete asset model with ID:{}.",
exception.getMessage());
                }
            });
    }

    /**
    * Creates a new IoT SiteWise portal.
    *
    * @param portalName the name of the portal to create.
    * @param iamRole the IAM role ARN to use for the portal.
    * @param contactEmail the email address of the portal contact.
    * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal ID. The calling code
    *      can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps

```

```

    *          it available to the calling code as a {@link CompletionException}.
By calling
    *          {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<String> createPortalAsync(String portalName, String
iamRole, String contactEmail) {
        CreatePortalRequest createPortalRequest = CreatePortalRequest.builder()
            .portalName(portalName)
            .portalDescription("This is my custom IoT SiteWise portal.")
            .portalContactEmail(contactEmail)
            .roleArn(iamRole)
            .build();

        return getAsyncClient().createPortal(createPortalRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create portal: {} ",
exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.portalId();
            });
    }

/**
 * Deletes a portal.
 *
 * @param portalId the ID of the portal to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeletePortalResponse}. The calling code can attach
 *          callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
 *          {@link CompletableFuture#get()}.
 *          <p>
 *          If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *          it available to the calling code as a {@link CompletionException}.
By calling
 *          {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<DeletePortalResponse> deletePortalAsync(String
portalId) {

```

```

        DeletePortalRequest deletePortalRequest = DeletePortalRequest.builder()
            .portalId(portalId)
            .build();

        return getAsyncClient().deletePortal(deletePortalRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete portal with ID: {}. Error:
{}\"", portalId, exception.getCause().getMessage());
                }
            });
    }

    /**
     * Retrieves the asset model ID for the given asset model name.
     *
     * @param assetModelName the name of the asset model for the ID.
     * @return a {@link CompletableFuture} that represents a {@link String}
     result of the asset model ID or null if the
     *     asset model cannot be found. The calling code can attach
     callbacks, then handle the result or exception
     *     by calling {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
     *
     <p>
     *     If any completion stage in this method throws an exception, the
     method logs the exception cause and keeps
     *     it available to the calling code as a {@link CompletionException}.
     By calling
     *     {@link CompletionException#getCause()}, the calling code can
     access the original exception.
     */
    public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
    {
        ListAssetModelsRequest listAssetModelsRequest =
ListAssetModelsRequest.builder().build();
        return getAsyncClient().listAssetModels(listAssetModelsRequest)
            .handle((listAssetModelsResponse, exception) -> {
                if (exception != null) {
                    logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {

```

```

        if (assetModelSummary.name().equals(assetModelName)) {
            return assetModelSummary.id();
        }
    }
    return null;
});
}

/**
 * Retrieves a portal's description.
 *
 * @param portalId the ID of the portal to describe.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal's start URL
 *      (see: {@link DescribePortalResponse#portalStartUrl()}). The
calling code can attach callbacks, then handle the
 *      result or exception by calling {@link CompletableFuture#join()} or
{@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> describePortalAsync(String portalId) {
    DescribePortalRequest request = DescribePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().describePortal(request)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("An exception occurred retrieving the portal
description: {}", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.portalStartUrl();
        });
}

/**

```

```

    * Creates a new IoT SiteWise gateway.
    *
    * @param gatewayName The name of the gateway to create.
    * @param myThing      The name of the core device thing to associate with the
gateway.
    * @return a {@link CompletableFuture} that represents a {@link String}
result of the gateways ID. The calling code
    *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *         {@link CompletableFuture#get()}.
    *         <p>
    *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *         it available to the calling code as a {@link CompletionException}.
By calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
        GreengrassV2 gg = GreengrassV2.builder()
            .coreDeviceThingName(myThing)
            .build();

        GatewayPlatform platform = GatewayPlatform.builder()
            .greengrassV2(gg)
            .build();

        Map<String, String> tag = new HashMap<>();
        tag.put("Environment", "Production");

        CreateGatewayRequest createGatewayRequest =
CreateGatewayRequest.builder()
            .gatewayName(gatewayName)
            .gatewayPlatform(platform)
            .tags(tag)
            .build();

        return getAsyncClient().createGateway(createGatewayRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Error creating the gateway.");
                    throw (CompletionException) exception;
                }
            })
    }

```



```

        logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
        return response.gatewayId();
    });
}

/**
 * Deletes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
 *      can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
    DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
}

/**
 * Describes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to describe.

```

```

    * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
    *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *         {@link CompletableFuture#get()}.
    *         <p>
    *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *         it available to the calling code as a {@link CompletionException}.
By calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
        DescribeGatewayRequest request = DescribeGatewayRequest.builder()
            .gatewayId(gatewayId)
            .build();

        return getAsyncClient().describeGateway(request)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
                }
            });
    }

    private static Map<String, Double> generateSampleData() {
        Map<String, Double> data = new HashMap<>();
        data.put("Temperature", 23.5);
        data.put("Humidity", 65.0);
        return data;
    }
}

```

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
  //} from "@aws-doc-sdk-examples/lib/scenario/index.js";
} from "../../libs/scenario/index.js";
import {
  IoTSiteWiseClient,
  CreateAssetModelCommand,
  CreateAssetCommand,
  ListAssetModelPropertiesCommand,
  BatchPutAssetPropertyValueCommand,
  GetAssetPropertyValueCommand,
  CreatePortalCommand,
  DescribePortalCommand,
  CreateGatewayCommand,
  DescribeGatewayCommand,
  DeletePortalCommand,
  DeleteGatewayCommand,
  DeleteAssetCommand,
  DeleteAssetModelCommand,
  DescribeAssetModelCommand,
} from "@aws-sdk/client-iotsitewise";
import {
  CloudFormationClient,
  CreateStackCommand,
  DeleteStackCommand,
  DescribeStacksCommand,
  waitUntilStackExists,
  waitUntilStackCreateComplete,
  waitUntilStackDeleteComplete,
```

```

} from "@aws-sdk/client-cloudformation";
import { wait } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { parseArgs } from "node:util";
import { readFileSync } from "node:fs";
import { fileURLToPath } from "node:url";
import { dirname } from "node:path";

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);
const stackName = "SiteWiseBasicsStack";

/**
 * @typedef {{
 *   iotSiteWiseClient: import('@aws-sdk/client-iotsitewise').IotSiteWiseClient,
 *   cloudFormationClient: import('@aws-sdk/client-cloudformation').CloudFormationClient,
 *   stackName,
 *   stack,
 *   askToDeleteResources: true,
 *   asset: {assetName: "MyAsset1"},
 *   assetModel: {assetModelName: "MyAssetModel1"},
 *   portal: {portalName: "MyPortal1"},
 *   gateway: {gatewayName: "MyGateway1"},
 *   propertyIds: [],
 *   contactEmail: "user@mydomain.com",
 *   thing: "MyThing1",
 *   sampleData: { temperature: 23.5, humidity: 65.0}
 * }} State
 */

/**
 * Used repeatedly to have the user press enter.
 * @type {ScenarioInput}
 */
const pressEnter = new ScenarioInput("continue", "Press Enter to continue", {
  type: "confirm",
});

const greet = new ScenarioOutput(
  "greet",
  `AWS IoT SiteWise is a fully managed industrial software-as-a-service (SaaS)
  that makes it easy to collect, store, organize, and monitor data from industrial
  equipment and processes. It is designed to help industrial and manufacturing

```

organizations collect data from their equipment and processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide range of industrial equipment and systems, including programmable logic controllers (PLCs), sensors, and other industrial devices. It can collect data from these devices and organize it into a unified data model, making it easier to analyze and gain insights from the data. AWS IoT SiteWise also provides tools for visualizing the data, setting up alarms and alerts, and generating reports. Another key feature of AWS IoT SiteWise is its ability to scale to handle large volumes of data. It can collect and store data from thousands of devices and process millions of data points per second, making it suitable for large-scale industrial operations. Additionally, AWS IoT SiteWise is designed to be secure and compliant, with features like role-based access controls, data encryption, and integration with other AWS services for additional security and compliance features.

Let's get started...`,

```
{ header: true },  
);
```

```
const displayBuildCloudFormationStack = new ScenarioOutput(  
  "displayBuildCloudFormationStack",  
  "This scenario uses AWS CloudFormation to create an IAM role that is required  
  for this scenario. The stack will now be deployed.",  
);
```

```
const sdkBuildCloudFormationStack = new ScenarioAction(  
  "sdkBuildCloudFormationStack",  
  async (** @type {State} */ state) => {  
    try {  
      const data = readFileSync(  
        `${__dirname}/../../../../../resources/cfn/iotsitewise_basics/SitewiseRoles-  
template.yml`,  
        "utf8",  
      );  
      await state.cloudFormationClient.send(  
        new CreateStackCommand({  
          StackName: stackName,  
          TemplateBody: data,  
          Capabilities: ["CAPABILITY_IAM"],  
        })),  
      );  
      await waitUntilStackExists(  
        { client: state.cloudFormationClient },
```

```

    { StackName: stackName },
  );
  await waitUntilStackCreateComplete(
    { client: state.cloudFormationClient },
    { StackName: stackName },
  );
  const stack = await state.cloudFormationClient.send(
    new DescribeStacksCommand({
      StackName: stackName,
    }),
  );
  state.stack = stack.Stacks[0].Outputs[0];
  console.log(`The ARN of the IAM role is ${state.stack.OutputValue}`);
} catch (caught) {
  console.error(caught.message);
  throw caught;
}
},
);

```

```

const displayCreateAWSSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSSiteWiseAssetModel",
  `1. Create an AWS SiteWise Asset Model

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and properties of each asset.

This scenario creates two asset model properties: temperature and humidity.`,

```

);

```

```

const sdkCreateAWSSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSSiteWiseAssetModel",
  async (/** @type {State} */ state) => {
    let assetModelResponse;
    try {
      assetModelResponse = await state.iotSiteWiseClient.send(
        new CreateAssetModelCommand({
          assetModelName: state.assetModel.assetModelName,
          assetModelProperties: [
            {
              name: "Temperature",
              dataType: "DOUBLE",
              type: {

```

```

        measurement: {},
      },
    },
    {
      name: "Humidity",
      dataType: "DOUBLE",
      type: {
        measurement: {},
      },
    },
  ],
  )),
);
state.assetModel.assetModelId = assetModelResponse.assetModelId;
console.log(
  `Asset Model successfully created. Asset Model ID:
  ${state.assetModel.assetModelId}`,
);
} catch (caught) {
  if (caught.name === "ResourceAlreadyExistsException") {
    console.log(
      `The Asset Model ${state.assetModel.assetModelName} already exists.`,
    );
    throw caught;
  }
  console.error(`${caught.message}`);
  throw caught;
}
},
);

const displayCreateAWSIoTSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSIoTSiteWiseAssetModel",
  `2. Create an AWS IoT SiteWise Asset
  The IoT SiteWise model that we just created defines the structure and metadata
  for your physical assets. Now we create an asset from the asset model.

  Let's wait 30 seconds for the asset to be ready.`,
);

const waitThirtySeconds = new ScenarioAction("waitThirtySeconds", async () => {
  await wait(30); // wait 30 seconds
  console.log("Time's up! Let's check the asset's status.");
});

```

```

const sdkCreateAWSIoTSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSIoTSiteWiseAssetModel",
  async (** @type {State} */ state) => {
    try {
      const assetResponse = await state.iotSiteWiseClient.send(
        new CreateAssetCommand({
          assetModelId: state.assetModel.assetModelId,
          assetName: state.asset.assetName,
        })),
      );
      state.asset.assetId = assetResponse.assetId;
      console.log(`Asset created with ID: ${state.asset.assetId}`);
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(
          `The Asset ${state.assetModel.assetModelName} was not found.`
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);

```

```

const displayRetrievePropertyId = new ScenarioOutput(
  "displayRetrievePropertyId",
  `3. Retrieve the property ID values

```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.`

```

);

```

```

const sdkRetrievePropertyId = new ScenarioAction(
  "sdkRetrievePropertyId",
  async (state) => {
    try {
      const retrieveResponse = await state.iotSiteWiseClient.send(
        new ListAssetModelPropertiesCommand({
          assetModelId: state.assetModel.assetModelId,
        })),
      );
    }
  }
);

```



```

    for (const retrieveResponseKey in
retrieveResponse.assetModelPropertySummaries) {
        if (
            retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
                .name === "Humidity"
        ) {
            state.propertyIds.Humidity =
                retrieveResponse.assetModelPropertySummaries[
                    retrieveResponseKey
                ].id;
        }
        if (
            retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
                .name === "Temperature"
        ) {
            state.propertyIds.Temperature =
                retrieveResponse.assetModelPropertySummaries[
                    retrieveResponseKey
                ].id;
        }
    }
    console.log(`The Humidity propertyId is ${state.propertyIds.Humidity}`);
    console.log(
        `The Temperature propertyId is ${state.propertyIds.Temperature}`,
    );
} catch (caught) {
    if (caught.name === "IoTSiteWiseException") {
        console.log(
            `There was a problem retrieving the properties: ${caught.message}`,
        );
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

```

```

const displaySendDataToIoTSiteWiseAsset = new ScenarioOutput(
    "displaySendDataToIoTSiteWiseAsset",
    `4. Send data to an AWS IoT SiteWise Asset

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a

centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.`,

);

```
const sdkSendDataToIoTSiteWiseAsset = new ScenarioAction(
  "sdkSendDataToIoTSiteWiseAsset",
  async (state) => {
    try {
      const sendResponse = await state.iotSiteWiseClient.send(
        new BatchPutAssetPropertyValueCommand({
          entries: [
            {
              entryId: "entry-3",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Humidity,
              propertyValues: [
                {
                  value: {
                    doubleValue: state.sampleData.humidity,
                  },
                  timestamp: {
                    timeInSeconds: Math.floor(Date.now() / 1000),
                  },
                },
              ],
            },
            {
              entryId: "entry-4",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Temperature,
              propertyValues: [
                {
                  value: {
                    doubleValue: state.sampleData.temperature,
                  },
                  timestamp: {
                    timeInSeconds: Math.floor(Date.now() / 1000),
                  },
                },
              ],
            },
          ],
        })
      );
    } catch (error) {
      console.error(error);
    }
  },
);
```

```

    ],
  )),
);
console.log("The data was sent successfully.");
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(`The Asset ${state.asset.assetName} was not found.`);
    throw caught;
  }
  console.error(`${caught.message}`);
  throw caught;
}
},
);

```

```

const displayRetrieveValueOfIoTSiteWiseAsset = new ScenarioOutput(
  "displayRetrieveValueOfIoTSiteWiseAsset",
  `5. Retrieve the value of the IoT SiteWise Asset property

```

IoT SiteWise is an AWS service that allows you to collect, process, and analyze industrial data from connected equipment and sensors. One of the key benefits of reading an IoT SiteWise property is the ability to gain valuable insights from your industrial data.`,

```

);

```

```

const sdkRetrieveValueOfIoTSiteWiseAsset = new ScenarioAction(
  "sdkRetrieveValueOfIoTSiteWiseAsset",
  async (** @type {State} */ state) => {
    try {
      const temperatureResponse = await state.iotSiteWiseClient.send(
        new GetAssetPropertyValueCommand({
          assetId: state.asset.assetId,
          propertyId: state.propertyIds.Temperature,
        }),
      );
      const humidityResponse = await state.iotSiteWiseClient.send(
        new GetAssetPropertyValueCommand({
          assetId: state.asset.assetId,
          propertyId: state.propertyIds.Humidity,
        }),
      );
      console.log(
        `The property value for Temperature is
        ${temperatureResponse.propertyValue.value.doubleValue}`,
      );
    } catch (error) {
      console.error(error);
    }
  },
);

```

```

    );
    console.log(
      `The property value for Humidity is
    ${humidityResponse.propertyValue.value.doubleValue}`,
    );
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

```

```

const displayCreateIoTSiteWisePortal = new ScenarioOutput(
  "displayCreateIoTSiteWisePortal",
  `6. Create an IoT SiteWise Portal

```

An IoT SiteWise Portal allows you to aggregate data from multiple industrial sources, such as sensors, equipment, and control systems, into a centralized platform.`,

```

);

```

```

const sdkCreateIoTSiteWisePortal = new ScenarioAction(
  "sdkCreateIoTSiteWisePortal",
  async (** @type {State} */ state) => {
    try {
      const createPortalResponse = await state.iotSiteWiseClient.send(
        new CreatePortalCommand({
          portalName: state.portal.portalName,
          portalContactEmail: state.contactEmail,
          roleArn: state.stack.OutputValue,
        }),
      );
      state.portal = { ...state.portal, ...createPortalResponse };
      await wait(5); // Allow the portal to properly propagate.
      console.log(
        `Portal created successfully. Portal ID
    ${createPortalResponse.portalId}`,
      );
    } catch (caught) {
      if (caught.name === "IoTSiteWiseException") {

```

```

        console.log(
            `There was a problem creating the Portal: ${caught.message}.`,
        );
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

```

```

const displayDescribePortal = new ScenarioOutput(
    "displayDescribePortal",
    `7. Describe the Portal

```

In this step, we get a description of the portal and display the portal URL.`

```

);

```

```

const sdkDescribePortal = new ScenarioAction(
    "sdkDescribePortal",
    async (** @type {State} */ state) => {
        try {
            const describePortalResponse = await state.iotSiteWiseClient.send(
                new DescribePortalCommand({
                    portalId: state.portal.portalId,
                }),
            );
            console.log(`Portal URL: ${describePortalResponse.portalStartUrl}`);
        } catch (caught) {
            if (caught.name === "ResourceNotFoundException") {
                console.log(`The Portal ${state.portal.portalName} was not found.`);
                throw caught;
            }
            console.error(`${caught.message}`);
            throw caught;
        }
    },
);

```

```

const displayCreateIoTSiteWiseGateway = new ScenarioOutput(
    "displayCreateIoTSiteWiseGateway",
    `8. Create an IoT SiteWise Gateway

```

IoT SiteWise Gateway serves as the bridge between industrial equipment, sensors, and the cloud-based IoT SiteWise service. It is responsible for securely collecting, processing, and transmitting data from various industrial assets to the IoT SiteWise platform, enabling real-time monitoring, analysis, and optimization of industrial operations.`,

);

```
const sdkCreateIoTSiteWiseGateway = new ScenarioAction(
  "sdkCreateIoTSiteWiseGateway",
  async (/** @type {State} */ state) => {
    try {
      const createGatewayResponse = await state.iotSiteWiseClient.send(
        new CreateGatewayCommand({
          gatewayName: state.gateway.gatewayName,
          gatewayPlatform: {
            greengrassV2: {
              coreDeviceThingName: state.thing,
            },
          },
        })),
      );
      console.log(
        `Gateway creation completed successfully. ID is
        ${createGatewayResponse.gatewayId}`,
      );
      state.gateway.gatewayId = createGatewayResponse.gatewayId;
    } catch (caught) {
      if (caught.name === "IoTSiteWiseException") {
        console.log(
          `There was a problem creating the gateway: ${caught.message}.`,
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);
```

```
const displayDescribeIoTSiteWiseGateway = new ScenarioOutput(
  "displayDescribeIoTSiteWiseGateway",
  "9. Describe the IoT SiteWise Gateway",
);
```

```

const sdkDescribeIoTSiteWiseGateway = new ScenarioAction(
  "sdkDescribeIoTSiteWiseGateway",
  async (** @type {State} */ state) => {
    try {
      const describeGatewayResponse = await state.iotSiteWiseClient.send(
        new DescribeGatewayCommand({
          gatewayId: state.gateway.gatewayId,
        })),
      );
      console.log("Gateway creation completed successfully.");
      console.log(`Gateway Name: ${describeGatewayResponse.gatewayName}`);
      console.log(`Gateway ARN: ${describeGatewayResponse.gatewayArn}`);
      console.log(
        `Gateway Platform:
${Object.keys(describeGatewayResponse.gatewayPlatform)}`,
      );
      console.log(
        `Gateway Creation Date: ${describeGatewayResponse.creationDate}`,
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);

const askToDeleteResources = new ScenarioInput(
  "askToDeleteResources",
  `10. Delete the AWS IoT SiteWise Assets

Before you can delete the Asset Model, you must delete the assets.`,
  { type: "confirm" },
);

const displayConfirmDeleteResources = new ScenarioAction(
  "displayConfirmDeleteResources",
  async (** @type {State} */ state) => {
    if (state.askToDeleteResources) {
      return "You selected to delete the SiteWise assets.";
    }
  }
);

```

```
        return "The resources will not be deleted. Please delete them manually to
        avoid charges.";
    },
);

const sdkDeleteResources = new ScenarioAction(
    "sdkDeleteResources",
    async (** @type {State} */ state) => {
        await wait(10); // Give the portal status time to catch up.
        try {
            await state.iotSiteWiseClient.send(
                new DeletePortalCommand({
                    portalId: state.portal.portalId,
                }),
            );
            console.log(
                `Portal ${state.portal.portalName} was deleted successfully.`,
            );
        } catch (caught) {
            if (caught.name === "ResourceNotFoundException") {
                console.log(`The Portal ${state.portal.portalName} was not found.`);
            } else {
                console.log(`When trying to delete the portal: ${caught.message}`);
            }
        }

        try {
            await state.iotSiteWiseClient.send(
                new DeleteGatewayCommand({
                    gatewayId: state.gateway.gatewayId,
                }),
            );
            console.log(
                `Gateway ${state.gateway.gatewayName} was deleted successfully.`,
            );
        } catch (caught) {
            if (caught.name === "ResourceNotFoundException") {
                console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
            } else {
                console.log(`When trying to delete the gateway: ${caught.message}`);
            }
        }

        try {
```



```
    await state.iotSiteWiseClient.send(
      new DeleteAssetCommand({
        assetId: state.asset.assetId,
      }),
    );
    await wait(5); // Allow the delete to finish.
    console.log(`Asset ${state.asset.assetName} was deleted successfully.`);
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
    } else {
      console.log(`When deleting the asset: ${caught.message}`);
    }
  }
}

await wait(30); // Allow asset deletion to finish.
try {
  await state.iotSiteWiseClient.send(
    new DeleteAssetModelCommand({
      assetModelId: state.assetModel.assetModelId,
    }),
  );
  console.log(
    `Asset Model ${state.assetModel.assetModelName} was deleted
successfully.`,
  );
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(
      `The Asset Model ${state.assetModel.assetModelName} was not found.`,
    );
  } else {
    console.log(`When deleting the asset model: ${caught.message}`);
  }
}

try {
  await state.cloudFormationClient.send(
    new DeleteStackCommand({
      StackName: stackName,
    }),
  );
  await waitUntilStackDeleteComplete(
    { client: state.cloudFormationClient },
```

```
        { StackName: stackName },
      );
      console.log("The stack was deleted successfully.");
    } catch (caught) {
      console.log(
        `${caught.message}. The stack was NOT deleted. Please clean up the
resources manually.`
      );
    }
  },
  { skipWhen: (/** @type {} */ state) => !state.askToDeleteResources },
);

const goodbye = new ScenarioOutput(
  "goodbye",
  "This concludes the IoT SiteWise Basics scenario for the AWS Javascript SDK v3.
Thank you!",
);

const myScenario = new Scenario(
  "IoTSiteWise Basics",
  [
    greet,
    pressEnter,
    displayBuildCloudFormationStack,
    sdkBuildCloudFormationStack,
    pressEnter,
    displayCreateAWSSiteWiseAssetModel,
    sdkCreateAWSSiteWiseAssetModel,
    displayCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    waitThirtySeconds,
    sdkCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    displayRetrievePropertyId,
    sdkRetrievePropertyId,
    pressEnter,
    displaySendDataToIoTSiteWiseAsset,
    sdkSendDataToIoTSiteWiseAsset,
    pressEnter,
    displayRetrieveValueOfIoTSiteWiseAsset,
    sdkRetrieveValueOfIoTSiteWiseAsset,
    pressEnter,
    displayCreateIoTSiteWisePortal,
```

```

    sdkCreateIoTSiteWisePortal,
    pressEnter,
    displayDescribePortal,
    sdkDescribePortal,
    pressEnter,
    displayCreateIoTSiteWiseGateway,
    sdkCreateIoTSiteWiseGateway,
    pressEnter,
    displayDescribeIoTSiteWiseGateway,
    sdkDescribeIoTSiteWiseGateway,
    pressEnter,
    askToDeleteResources,
    displayConfirmDeleteResources,
    sdkDeleteResources,
    goodbye,
  ],
  {
    iotSiteWiseClient: new IoTSiteWiseClient({}),
    cloudFormationClient: new CloudFormationClient({}),
    asset: { assetName: "MyAsset1" },
    assetModel: { assetModelName: "MyAssetModel1" },
    portal: { portalName: "MyPortal1" },
    gateway: { gatewayName: "MyGateway1" },
    propertyIds: [],
    contactEmail: "user@mydomain.com",
    thing: "MyThing1",
    sampleData: { temperature: 23.5, humidity: 65.0 },
  },
);

/** @type {{ stepHandlerOptions: StepHandlerOptions }} */
export const main = async (stepHandlerOptions) => {
  await myScenario.run(stepHandlerOptions);
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const { values } = parseArgs({
    options: {
      yes: {
        type: "boolean",
        short: "y",
      },
    },
  },
);

```

```
});  
main({ confirmAll: values.yes });  
}
```

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkap dan pelajari cara menyiapkan dan menjalankan di [Repositori Contoh Kode AWS](#).

Jalankan skenario interaktif di penggugah/prompt perintah.

```
class IoTSitewiseGettingStarted:  
    """  
    A scenario that demonstrates how to use Boto3 to manage IoT physical assets  
    using  
    the AWS IoT SiteWise.  
    """  
  
    def __init__(  
        self,  
        iot_sitewise_wrapper: IoTSitewiseWrapper,  
        cloud_formation_resource: ServiceResource,  
    ):  
        self.iot_sitewise_wrapper = iot_sitewise_wrapper  
        self.cloud_formation_resource = cloud_formation_resource  
        self.stack = None  
        self.asset_model_id = None  
        self.asset_id = None  
        self.portal_id = None  
        self.gateway_id = None  
  
    def run(self) -> None:  
        """  
        Runs the scenario.  
        """
```

```
print(
    """
AWS IoT SiteWise is a fully managed software-as-a-service (SaaS) that
makes it easy to collect, store, organize, and monitor data from industrial
equipment and processes.
It is designed to help industrial and manufacturing organizations collect data
from their equipment and
processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide
range of industrial
equipment and systems, including programmable logic controllers (PLCs), sensors,
and other
industrial devices. It can collect data from these devices and organize it into a
unified data model,
making it easier to analyze and gain insights from the data. AWS IoT SiteWise
also provides tools for
visualizing the data, setting up alarms and alerts, and generating reports.

Another key feature of AWS IoT SiteWise is its ability to scale to handle large
volumes of data.
It can collect and store data from thousands of devices and process millions of
data points per second,
making it suitable for large-scale industrial operations. Additionally, AWS IoT
SiteWise is designed
to be secure and compliant, with features like role-based access controls, data
encryption,
and integration with other AWS services for additional security and compliance
features.

Let's get started...
    """
)
press_enter_to_continue()
print_dashes()
print(f"")
print(
    f"Use AWS CloudFormation to create an IAM role that is required for
this scenario."
)
template_file = IoTSiteWiseGettingStarted.get_template_as_string()

self.stack = self.deploy_cloudformation_stack(
    "python-iot-sitewise-basics", template_file
```

```

    )
    outputs = self.stack.outputs
    iam_role = None

    for output in outputs:
        if output.get("OutputKey") == "SitewiseRoleArn":
            iam_role = output.get("OutputValue")

    if iam_role is None:
        error_string = f"Failed to retrieve iam_role from CloudFormation
stack."
        logger.error(error_string)
        raise ValueError(error_string)

    print(f"The ARN of the IAM role is {iam_role}")
    print_dashes()
    print_dashes()
    print(f"1. Create an AWS SiteWise Asset Model")
    print(
        """

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and values of each asset.

This scenario creates two asset model values: temperature and humidity.

```

    """
    )
    press_enter_to_continue()
    asset_model_name = "MyAssetModel1"
    temperature_property_name = "temperature"
    humidity_property_name = "humidity"
    try:
        properties = [
            {
                "name": temperature_property_name,
                "dataType": "DOUBLE",
                "type": {
                    "measurement": {},
                },
            },
        ],
    },

```

```

        {
            "name": humidity_property_name,
            "dataType": "DOUBLE",
            "type": {
                "measurement": {},
            },
        },
    ]
    self.asset_model_id = self.iot_sitewise_wrapper.create_asset_model(
        asset_model_name, properties
    )
    print(
        f"Asset Model successfully created. Asset Model ID:
{self.asset_model_id}. "
    )
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            self.asset_model_id =
self.get_model_id_for_model_name(asset_model_name)
            print(
                f"Asset Model {asset_model_name} already exists. Asset Model
ID: {self.asset_model_id}. "
            )
        else:
            raise

    press_enter_to_continue()
    print_dashes()
    print(f"2. Create an AWS IoT SiteWise Asset")
    print(
        """

```

The IoT SiteWise model that we just created defines the structure and metadata for your physical assets.

Now we create an asset from the asset model.

```

        """
    )
    press_enter_to_continue()

    self.asset_id = self.iot_sitewise_wrapper.create_asset(
        "MyAsset1", self.asset_model_id
    )

    print(f"Asset created with ID: {self.asset_id}")

```

```

    press_enter_to_continue()
    print_dashes()
    print_dashes()
    print(f"3. Retrieve the property ID values")
    print(
        """

```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.

```

        """
    )
    press_enter_to_continue()
    property_ids = self.iot_sitewise_wrapper.list_asset_model_properties(
        self.asset_model_id
    )
    humidity_property_id = None
    temperature_property_id = None
    for property_id in property_ids:
        if property_id.get("name") == humidity_property_name:
            humidity_property_id = property_id.get("id")
        elif property_id.get("name") == temperature_property_name:
            temperature_property_id = property_id.get("id")
    if humidity_property_id is None or temperature_property_id is None:
        error_string = f"Failed to retrieve property IDs from Asset Model."
        logger.error(error_string)
        raise ValueError(error_string)

    print(f"The Humidity property Id is {humidity_property_id}")
    print(f"The Temperature property Id is {temperature_property_id}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"4. Send data to an AWS IoT SiteWise Asset")
    print(
        """

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.


```

        """
    )
    press_enter_to_continue()

    values = [
        {
            "propertyId": humidity_property_id,
            "valueType": "doubleValue",
            "value": 65.0,
        },
        {
            "propertyId": temperature_property_id,
            "valueType": "doubleValue",
            "value": 23.5,
        },
    ]
    self.iot_sitewise_wrapper.batch_put_asset_property_value(self.asset_id,
values)
    print(f>Data sent successfully.")

    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"5. Retrieve the value of the IoT SiteWise Asset property")
    print(
        """
IoT SiteWise is an AWS service that allows you to collect, process, and analyze
industrial data
from connected equipment and sensors. One of the key benefits of reading an IoT
SiteWise property
is the ability to gain valuable insights from your industrial data.

        """
    )
    press_enter_to_continue()

    property_value = self.iot_sitewise_wrapper.get_asset_property_value(
        self.asset_id, temperature_property_id
    )
    print(f"The property name is '{temperature_property_name}'.")

    print(

```

```

        f"The value of this property is: {property_value['value']}
['doubleValue']}"
    )
    press_enter_to_continue()

    property_value = self.iot_sitewise_wrapper.get_asset_property_value(
        self.asset_id, humidity_property_id
    )
    print(f"The property name is '{humidity_property_name}'.")
    print(
        f"The value of this property is: {property_value['value']}
['doubleValue']}"
    )
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"6. Create an IoT SiteWise Portal")
    print(
        """
An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources,
such as sensors, equipment, and control systems, into a centralized platform.
        """
    )

    press_enter_to_continue()
    contact_email = q.ask("Enter a contact email for the portal:",
q.non_empty)
    print("Creating the portal. The portal may take a while to become
active.")
    self.portal_id = self.iot_sitewise_wrapper.create_portal(
        "MyPortal1", iam_role, contact_email
    )
    print(f"Portal created successfully. Portal ID {self.portal_id}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"7. Describe the Portal")
    print(
        """
In this step, we get a description of the portal and display the portal URL.
        """

```

```

    )
    press_enter_to_continue()
    portal_description =
self.iot_sitewise_wrapper.describe_portal(self.portal_id)
    print(f"Portal URL: {portal_description['portalStartUrl']}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"8. Create an IoT SiteWise Gateway")
    press_enter_to_continue()
    self.gateway_id = self.iot_sitewise_wrapper.create_gateway(
        "MyGateway1", "MyThing1"
    )
    print(f"Gateway creation completed successfully. id is
{self.gateway_id}")
    print_dashes()
    print_dashes()
    print(f"9. Describe the IoT SiteWise Gateway")
    press_enter_to_continue()

    gateway_description = self.iot_sitewise_wrapper.describe_gateway(
        self.gateway_id
    )
    print(f"Gateway Name: {gateway_description['gatewayName']}")
    print(f"Gateway ARN: {gateway_description['gatewayArn']}")
    print(f"Gateway Platform:\n{gateway_description['gatewayPlatform']}")
    print(f"Gateway Creation Date: {gateway_description['gatewayArn']}")
    print_dashes()
    print_dashes()

    print(f"10. Delete the AWS IoT SiteWise Assets")
    if q.ask("Would you like to delete the IoT SiteWise Assets? (y/n)",
q.is_yesno):
        self.cleanup()
    else:
        print(f"The resources will not be deleted.")
    print_dashes()
    print_dashes()
    print(f"This concludes the AWS IoT SiteWise Scenario")

def cleanup(self) -> None:
    """
    Deletes the CloudFormation stack and the resources created for the demo.

```

```

"""

if self.gateway_id is not None:
    self.iot_sitewise_wrapper.delete_gateway(self.gateway_id)
    print(f"Deleted gateway with id {self.gateway_id}.")
    self.gateway_id = None
if self.portal_id is not None:
    self.iot_sitewise_wrapper.delete_portal(self.portal_id)
    print(f"Deleted portal with id {self.portal_id}.")
    self.portal_id = None
if self.asset_id is not None:
    self.iot_sitewise_wrapper.delete_asset(self.asset_id)
    print(f"Deleted asset with id {self.asset_id}.")
    self.iot_sitewise_wrapper.wait_asset_deleted(self.asset_id)
    self.asset_id = None
if self.asset_model_id is not None:
    self.iot_sitewise_wrapper.delete_asset_model(self.asset_model_id)
    print(f"Deleted asset model with id {self.asset_model_id}.")
    self.asset_model_id = None
if self.stack is not None:
    stack = self.stack
    self.stack = None
    self.destroy_cloudformation_stack(stack)

def deploy_cloudformation_stack(
    self, stack_name: str, cfn_template: str
) -> ServiceResource:
    """
    Deploys prerequisite resources used by the scenario. The resources are
    defined in the associated `SitewiseRoles-template.yaml` AWS
    CloudFormation script and are deployed
    as a CloudFormation stack, so they can be easily managed and destroyed.

    :param stack_name: The name of the CloudFormation stack.
    :param cfn_template: The CloudFormation template as a string.
    :return: The CloudFormation stack resource.
    """
    print(f"Deploying CloudFormation stack: {stack_name}.")
    stack = self.cloud_formation_resource.create_stack(
        StackName=stack_name,
        TemplateBody=cfn_template,
        Capabilities=["CAPABILITY_NAMED_IAM"],
    )
    print(f"CloudFormation stack creation started: {stack_name}")

```

```

        print("Waiting for CloudFormation stack creation to complete...")
        waiter = self.cloud_formation_resource.meta.client.get_waiter(
            "stack_create_complete"
        )
        waiter.wait(StackName=stack.name)
        stack.load()
        print("CloudFormation stack creation complete.")

    return stack

def destroy_cloudformation_stack(self, stack: ServiceResource) -> None:
    """
    Destroys the resources managed by the CloudFormation stack, and the
    CloudFormation
    stack itself.

    :param stack: The CloudFormation stack that manages the example
    resources.
    """
    print(
        f"CloudFormation stack '{stack.name}' is being deleted. This may take
    a few minutes."
    )
    stack.delete()
    waiter = self.cloud_formation_resource.meta.client.get_waiter(
        "stack_delete_complete"
    )
    waiter.wait(StackName=stack.name)
    print(f"CloudFormation stack '{stack.name}' has been deleted.")

    @staticmethod
    def get_template_as_string() -> str:
        """
        Returns a string containing this scenario's CloudFormation template.
        """
        template_file_path = os.path.join(script_dir, "SitewiseRoles-
    template.yaml")
        file = open(template_file_path, "r")
        return file.read()

    def get_model_id_for_model_name(self, model_name: str) -> str:
        """
        Returns the model ID for the given model name.

```

```

:param model_name: The name of the model.
:return: The model ID.
"""
model_id = None
asset_models = self.iot_sitewise_wrapper.list_asset_models()
for asset_model in asset_models:
    if asset_model["name"] == model_name:
        model_id = asset_model["id"]
        break
return model_id

```

Kelas `IoTSitewiseWrapper` yang membungkus tindakan AWS IoT SiteWise .

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

```

```

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )
        raise

def create_asset(self, asset_name: str, asset_model_id: str) -> str:
    """
    Creates an AWS IoT SiteWise Asset.

    :param asset_name: The name of the asset to create.
    :param asset_model_id: The ID of the asset model to associate with the
asset.
    :return: The ID of the created asset.
    """
    try:

```

```

        response = self.iotsitewise_client.create_asset(
            assetName=asset_name, assetModelId=asset_model_id
        )
        asset_id = response["assetId"]
        waiter = self.iotsitewise_client.get_waiter("asset_active")
        waiter.wait(assetId=asset_id)
        return asset_id
    except ClientError as err:
        if err.response["Error"] == "ResourceNotFoundException":
            logger.error("Asset model %s does not exist.", asset_model_id)
        else:
            logger.error(
                "Error creating asset %s. Here's why %s",
                asset_name,
                err.response["Error"]["Message"],
            )
            raise

def list_asset_models(self) -> List[Dict[str, Any]]:
    """
    Lists all AWS IoT SiteWise Asset Models.

    :return: A list of dictionaries containing information about each asset
    model.

    """
    try:
        asset_models = []
        paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
        pages = paginator.paginate()
        for page in pages:
            asset_models.extend(page["assetModelSummaries"])
        return asset_models
    except ClientError as err:
        logger.error(
            "Error listing asset models. Here's why %s",
            err.response["Error"]["Message"],
        )
        raise

```



```

def list_asset_model_properties(self, asset_model_id: str) -> List[Dict[str,
Any]]:
    """
    Lists all AWS IoT SiteWise Asset Model Properties.

    :param asset_model_id: The ID of the asset model to list values for.
    :return: A list of dictionaries containing information about each asset
model property.
    """
    try:
        asset_model_properties = []
        paginator = self.iotsitewise_client.get_paginator(
            "list_asset_model_properties"
        )
        pages = paginator.paginate(assetModelId=asset_model_id)
        for page in pages:
            asset_model_properties.extend(page["assetModelPropertySummaries"])
        return asset_model_properties
    except ClientError as err:
        logger.error(
            "Error listing asset model values. Here's why %s",
            err.response["Error"]["Message"],
        )
        raise

def batch_put_asset_property_value(
    self, asset_id: str, values: List[Dict[str, str]]
) -> None:
    """
    Sends data to an AWS IoT SiteWise Asset.

    :param asset_id: The asset ID.
    :param values: A list of dictionaries containing the values in the form
        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
         value : the_value}.
    """
    try:
        entries = self.properties_to_values(asset_id, values)

        self.iotsitewise_client.batch_put_asset_property_value(entries=entries)

```

```

except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Asset %s does not exist.", asset_id)
    else:
        logger.error(
            "Error sending data to asset. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise

def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form
        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
                 value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
    batch_put_asset_property_value.
    """
    entries = []
    for value in values:
        epoch_ns = time.time_ns()
        self.entry_id += 1
        if value["valueType"] == "stringValue":
            property_value = {"stringValue": value["value"]}
        elif value["valueType"] == "integerValue":
            property_value = {"integerValue": value["value"]}
        elif value["valueType"] == "booleanValue":
            property_value = {"booleanValue": value["value"]}
        elif value["valueType"] == "doubleValue":
            property_value = {"doubleValue": value["value"]}
        else:
            raise ValueError("Invalid valueType: %s", value["valueType"])
        entry = {
            "entryId": f"{self.entry_id}",
            "assetId": asset_id,
            "propertyId": value["propertyId"],

```

```

        "propertyValues": [
            {
                "value": property_value,
                "timestamp": {
                    "timeInSeconds": int(epoch_ns / 1000000000),
                    "offsetInNanos": epoch_ns % 1000000000,
                },
            }
        ],
    }
    entries.append(entry)
return entries

def get_asset_property_value(
    self, asset_id: str, property_id: str
) -> Dict[str, Any]:
    """
    Gets the value of an AWS IoT SiteWise Asset Property.

    :param asset_id: The ID of the asset.
    :param property_id: The ID of the property.
    :return: A dictionary containing the value of the property.
    """
    try:
        response = self.iotsitewise_client.get_asset_property_value(
            assetId=asset_id, propertyId=property_id
        )
        return response["propertyValue"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error(
                "Asset %s or property %s does not exist.", asset_id,
property_id
            )
        else:
            logger.error(
                "Error getting asset property value. Here's why %s",
                err.response["Error"]["Message"],
            )
        raise

def create_portal(

```

```

        self, portal_name: str, iam_role_arn: str, portal_contact_email: str
    ) -> str:
        """
        Creates an AWS IoT SiteWise Portal.

        :param portal_name: The name of the portal to create.
        :param iam_role_arn: The ARN of an IAM role.
        :param portal_contact_email: The contact email of the portal.
        :return: The ID of the created portal.
        """
        try:
            response = self.iotsitewise_client.create_portal(
                portalName=portal_name,
                roleArn=iam_role_arn,
                portalContactEmail=portal_contact_email,
            )
            portal_id = response["portalId"]
            waiter = self.iotsitewise_client.get_waiter("portal_active")
            waiter.wait(portalId=portal_id, WaiterConfig={"MaxAttempts": 40})
            return portal_id
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
                logger.error("Portal %s already exists.", portal_name)
            else:
                logger.error(
                    "Error creating portal %s. Here's why %s",
                    portal_name,
                    err.response["Error"]["Message"],
                )
            raise

def describe_portal(self, portal_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to describe.
    :return: A dictionary containing information about the portal.
    """
    try:
        response =
self.iotsitewise_client.describe_portal(portalId=portal_id)
        return response
    except ClientError as err:

```

```

        logger.error(
            "Error describing portal %s. Here's why %s",
            portal_id,
            err.response["Error"]["Message"],
        )
        raise

def create_gateway(self, gateway_name: str, my_thing: str) -> str:
    """
    Creates an AWS IoT SiteWise Gateway.

    :param gateway_name: The name of the gateway to create.
    :param my_thing: The core device thing name.
    :return: The ID of the created gateway.
    """
    try:
        response = self.iotsitewise_client.create_gateway(
            gatewayName=gateway_name,
            gatewayPlatform={
                "greengrassV2": {"coreDeviceThingName": my_thing},
            },
            tags={"Environment": "Production"},
        )
        gateway_id = response["gatewayId"]
        return gateway_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Gateway %s already exists.", gateway_name)
        else:
            logger.error(
                "Error creating gateway %s. Here's why %s",
                gateway_name,
                err.response["Error"]["Message"],
            )
            raise

def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
    """

```

```

    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_portal(self, portal_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to delete.
    """
    try:

```

```

        self.iotsitewise_client.delete_portal(portalId=portal_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Portal %s does not exist.", portal_id)
        else:
            logger.error(
                "Error deleting portal %s. Here's why %s",
                portal_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

def delete_asset_model(self, asset_model_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset Model.

    :param asset_model_id: The ID of the asset model to delete.
    """
    try:
        self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset model %s. Here's why %s",
            asset_model_id,
            err.response["Error"]["Message"],
        )

```

```
)
raise

def wait_asset_deleted(self, asset_id: str) -> None:
    """
    Waits for an AWS IoT SiteWise Asset to be deleted.

    :param asset_id: The ID of the asset to wait for.
    """
    try:
        waiter = self.iotsitewise_client.get_waiter("asset_not_exists")
        waiter.wait(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error waiting for asset %s to be deleted. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tindakan untuk AWS IoT SiteWise menggunakan AWS SDKs

Contoh kode berikut menunjukkan bagaimana melakukan AWS IoT SiteWise tindakan individu dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat [Referensi AWS IoT SiteWise API](#).

Contoh

- [Gunakan BatchPutAssetPropertyValue dengan AWS SDK atau CLI](#)
- [Gunakan CreateAsset dengan AWS SDK atau CLI](#)

- [Gunakan CreateAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan CreateGateway dengan AWS SDK atau CLI](#)
- [Gunakan CreatePortal dengan AWS SDK atau CLI](#)
- [Gunakan DeleteAsset dengan AWS SDK atau CLI](#)
- [Gunakan DeleteAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan DeleteGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeletePortal dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAssetModel dengan AWS SDK atau CLI](#)
- [Gunakan DescribeGateway dengan AWS SDK atau CLI](#)
- [Gunakan DescribePortal dengan AWS SDK atau CLI](#)
- [Gunakan GetAssetPropertyValue dengan AWS SDK atau CLI](#)
- [Gunakan ListAssetModels dengan AWS SDK atau CLI](#)

Gunakan **BatchPutAssetPropertyValue** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `BatchPutAssetPropertyValue`.

CLI

AWS CLI

Untuk mengirim data ke properti aset

`batch-put-asset-property-value` Contoh berikut mengirimkan data daya dan suhu ke properti aset yang diidentifikasi oleh alias properti.

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

Isi dari `batch-put-asset-property-value.json`:

```
{
  "entries": [
    {
      "entryId": "1575691200-company-windfarm-3-turbine-7-power",
```

```

    "propertyAlias": "company-windfarm-3-turbine-7-power",
    "propertyValues": [
      {
        "value": {
          "doubleValue": 4.92
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        },
        "quality": "GOOD"
      }
    ],
  },
  {
    "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
    "propertyAlias": "company-windfarm-3-turbine-7-temperature",
    "propertyValues": [
      {
        "value": {
          "integerValue": 38
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        }
      }
    ]
  }
]
}

```

Output:

```

{
  "errorEntries": []
}

```

Untuk informasi selengkapnya, lihat [Menyerap data menggunakan API AWS SiteWise IoT](#) di Panduan Pengguna AWS IoT SiteWise .

- Untuk detail API, lihat [BatchPutAssetPropertyValue](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Sends data to the SiteWise service.
 *
 * @param assetId      the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.
 * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
 *      calling code can attach callbacks, then handle the result or
exception by calling
 *      {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *      available to the calling code as a {@link CompletionException}. By
calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();
```

```

        BatchPutAssetPropertyValueRequest request =
BatchPutAssetPropertyValueRequest.builder()
    .entries(Arrays.asList(
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-3")
            .assetId(assetId)
            .propertyId(tempPropertyId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Temperature"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build(),
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-4")
            .assetId(assetId)
            .propertyId(humidityPropId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Humidity"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build()
    ))
    .build();

return getAsyncClient().batchPutAssetPropertyValue(request)
    .whenComplete((response, exception) -> {
        if (exception != null) {
            logger.error("An exception occurred: {}",
exception.getCause().getMessage());
        }
    });
}

```

- Untuk detail API, lihat [BatchPutAssetPropertyValue](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  BatchPutAssetPropertyValueCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Batch put asset property values.
 * @param {{ entries : array }}
 */
export const main = async ({ entries }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new BatchPutAssetPropertyValueCommand({
        entries: entries,
      }),
    );
    console.log("Asset properties batch put successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(`${caught.message}. A resource could not be found.`);
    } else {
      throw caught;
    }
  }
};
```

- Untuk detail API, lihat [BatchPutAssetPropertyValue](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def batch_put_asset_property_value(
        self, asset_id: str, values: List[Dict[str, str]]
    ) -> None:
        """
```

```

        Sends data to an AWS IoT SiteWise Asset.

        :param asset_id: The asset ID.
        :param values: A list of dictionaries containing the values in the form
                        {propertyId : property_id,
                         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
                                value : the_value}.
        """
        try:
            entries = self.properties_to_values(asset_id, values)

        self.iotsitewise_client.batch_put_asset_property_value(entries=entries)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error("Asset %s does not exist.", asset_id)
            else:
                logger.error(
                    "Error sending data to asset. Here's why %s",
                    err.response["Error"]["Message"],
                )
            raise

```

Fungsi pembantu untuk menghasilkan parameter entri dari daftar nilai.

```

def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form
                    {propertyId : property_id,
                     valueType : [stringValue|integerValue|doubleValue|
booleanValue],
                            value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
    batch_put_asset_property_value.
    """
    entries = []

```

```

for value in values:
    epoch_ns = time.time_ns()
    self.entry_id += 1
    if value["valueType"] == "stringValue":
        property_value = {"stringValue": value["value"]}
    elif value["valueType"] == "integerValue":
        property_value = {"integerValue": value["value"]}
    elif value["valueType"] == "booleanValue":
        property_value = {"booleanValue": value["value"]}
    elif value["valueType"] == "doubleValue":
        property_value = {"doubleValue": value["value"]}
    else:
        raise ValueError("Invalid valueType: %s", value["valueType"])
    entry = {
        "entryId": f"{self.entry_id}",
        "assetId": asset_id,
        "propertyId": value["propertyId"],
        "propertyValues": [
            {
                "value": property_value,
                "timestamp": {
                    "timeInSeconds": int(epoch_ns / 1000000000),
                    "offsetInNanos": epoch_ns % 1000000000,
                },
            },
        ],
    }
    entries.append(entry)
return entries

```

Berikut adalah contoh daftar nilai untuk diteruskan ke fungsi pembantu.

```

values = [
    {
        "propertyId": humidity_property_id,
        "valueType": "doubleValue",
        "value": 65.0,
    },
    {
        "propertyId": temperature_property_id,
        "valueType": "doubleValue",
    },
]

```



```
        "value": 23.5,
      },
    ]
  }
```

- Untuk detail API, lihat [BatchPutAssetPropertyValue](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateAsset** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateAsset`.

CLI

AWS CLI

Untuk membuat aset

`create-asset` Contoh berikut menciptakan aset turbin angin dari model aset turbin angin.

```
aws iotsitewise create-asset \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --asset-name "Wind Turbine 1"
```

Output:

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "assetStatus": {
    "state": "CREATING"
  }
}
```

Untuk informasi selengkapnya, lihat [Membuat aset](#) di SiteWise Panduan Pengguna AWS IoT.

- Untuk detail API, lihat [CreateAsset](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *      available to the calling code as a {@link CompletionException}. By
calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
    CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
        .assetModelId(assetModelId)
        .assetDescription("Created using the AWS SDK for Java")
        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
```

```

        logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
    }
    });
}

```

- Untuk detail API, lihat [CreateAsset](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
  CreateAssetCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetName, assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateAssetCommand({
        assetName: assetName, // The name to give the Asset.
        assetModelId: assetModelId, // The ID of the asset model from which to
        create the asset.
      }),
    );
    console.log("Asset created successfully.");
    return result;
  } catch (caught) {

```

```

    if (caught instanceof Error && caught.name === "ResourceNotFound") {
        console.warn(
            `${caught.message}. The asset model could not be found. Please check the
            asset model id.` ,
        );
    } else {
        throw caught;
    }
}
};

```

- Untuk detail API, lihat [CreateAsset](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """

```

```

        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def create_asset(self, asset_name: str, asset_model_id: str) -> str:
    """
    Creates an AWS IoT SiteWise Asset.

    :param asset_name: The name of the asset to create.
    :param asset_model_id: The ID of the asset model to associate with the
    asset.

    :return: The ID of the created asset.
    """
    try:
        response = self.iotsitewise_client.create_asset(
            assetName=asset_name, assetModelId=asset_model_id
        )
        asset_id = response["assetId"]
        waiter = self.iotsitewise_client.get_waiter("asset_active")
        waiter.wait(assetId=asset_id)
        return asset_id
    except ClientError as err:
        if err.response["Error"] == "ResourceNotFoundException":
            logger.error("Asset model %s does not exist.", asset_model_id)
        else:
            logger.error(
                "Error creating asset %s. Here's why %s",
                asset_name,
                err.response["Error"]["Message"],
            )
        raise

```

- Untuk detail API, lihat [CreateAsset](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateAssetModel** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateAssetModel`.

CLI

AWS CLI

Untuk membuat model aset

`create-asset-model` Contoh berikut menciptakan model aset yang mendefinisikan turbin angin dengan sifat-sifat berikut:

Nomor seri - Nomor seri turbin angin
Daya energi - Aliran data daya yang dihasilkan dari turbin angin
Suhu C - Aliran data suhu dari turbin angin di Celsius
Temperature F - Titik data suhu yang dipetakan dari Celcius ke Fahrenheit

```
aws iotsitewise create-asset-model \
  --cli-input-json file://create-wind-turbine-model.json
```

Isi dari `create-wind-turbine-model.json`:

```
{
  "assetModelName": "Wind Turbine Model",
  "assetModelDescription": "Represents a wind turbine",
  "assetModelProperties": [
    {
      "name": "Serial Number",
      "dataType": "STRING",
      "type": {
        "attribute": {}
      }
    },
    {
      "name": "Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "measurement": {}
      }
    }
  ]
}
```

```

    }
  },
  {
    "name": "Temperature C",
    "dataType": "DOUBLE",
    "unit": "Celsius",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "Temperature F",
    "dataType": "DOUBLE",
    "unit": "Fahrenheit",
    "type": {
      "transform": {
        "expression": "temp_c * 9 / 5 + 32",
        "variables": [
          {
            "name": "temp_c",
            "value": {
              "propertyId": "Temperature C"
            }
          }
        ]
      }
    }
  },
  {
    "name": "Total Generated Power",
    "dataType": "DOUBLE",
    "unit": "kW",
    "type": {
      "metric": {
        "expression": "sum(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "Generated Power"
            }
          }
        ]
      },
      "window": {

```

```

        "tumbling": {
            "interval": "1h"
        }
    }
}

```

Output:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Untuk informasi selengkapnya, lihat [Mendefinisikan model aset](#) di Panduan AWS Pengguna SiteWise IoT.

- Untuk detail API, lihat [CreateAssetModel](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.

```



```

    * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
    *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *         {@link CompletableFuture#get()}.
    *         <p>
    *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
    *         available to the calling code as a {@link CompletionException}. By
calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
        PropertyType humidity = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        PropertyType temperaturePropertyType = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
            .name("Temperature")
            .dataType(PropertyDataType.DOUBLE)
            .type(temperaturePropertyType)
            .build();

        AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
            .name("Humidity")
            .dataType(PropertyDataType.DOUBLE)
            .type(humidity)
            .build();

        CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
            .assetModelName(name)
            .assetModelDescription("This is my asset model")
            .assetModelProperties(temperatureProperty, humidityProperty)
            .build();

```

```

        return getAsyncClient().createAssetModel(createAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset model: {} ",
exception.getCause().getMessage());
                }
            });
    }
}

```

- Untuk detail API, lihat [CreateAssetModel](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    CreateAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset Model.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetModelName, assetModelId }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const result = await client.send(
            new CreateAssetModelCommand({
                assetModelName: assetModelName, // The name to give the Asset Model.
            }),
        );
        console.log("Asset model created successfully.");
        return result;
    }
}

```

```

    } catch (caught) {
      if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
        console.warn(
          `${caught.message}. There was a problem creating the asset model.`
        );
      } else {
        throw caught;
      }
    }
  };
};

```

- Untuk detail API, lihat [CreateAssetModel](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """

```

```

        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )
        raise

```

Berikut adalah contoh daftar properti untuk diteruskan ke fungsi.

```
properties = [
    {
        "name": temperature_property_name,
        "dataType": "DOUBLE",
        "type": {
            "measurement": {},
        },
    },
    {
        "name": humidity_property_name,
        "dataType": "DOUBLE",
        "type": {
            "measurement": {},
        },
    },
]
```

- Untuk detail API, lihat [CreateAssetModel](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateGateway`.

CLI

AWS CLI

Untuk membuat gateway

`create-gateway` Contoh berikut membuat gateway yang berjalan di AWS IoT Greengrass.

```
aws iotsitewise create-gateway \
    --gateway-name ExampleCorpGateway \
    --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}
```

Output:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

Untuk informasi selengkapnya, lihat [Mengonfigurasi gateway](#) di Panduan AWS Pengguna SiteWise IoT.

- Untuk detail API, lihat [CreateGateway](#) di Referensi AWS CLI Perintah.

Java**SDK untuk Java 2.x****Note**

Ada lebih banyak tentang GitHub. Temukan contoh selengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Creates a new IoT Sitewise gateway.
 *
 * @param gatewayName The name of the gateway to create.
 * @param myThing      The name of the core device thing to associate with the
gateway.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the gateways ID. The calling code
 *       can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *       {@link CompletableFuture#get()}.
 *       <p>
 *       If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *       it available to the calling code as a {@link CompletionException}.
By calling
 *       {@link CompletionException#getCause()}, the calling code can
access the original exception.
```

```
*/
public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
    GreengrassV2 gg = GreengrassV2.builder()
        .coreDeviceThingName(myThing)
        .build();

    GatewayPlatform platform = GatewayPlatform.builder()
        .greengrassV2(gg)
        .build();

    Map<String, String> tag = new HashMap<>();
    tag.put("Environment", "Production");

    CreateGatewayRequest createGatewayRequest =
    CreateGatewayRequest.builder()
        .gatewayName(gatewayName)
        .gatewayPlatform(platform)
        .tags(tag)
        .build();

    return getAsyncClient().createGateway(createGatewayRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Error creating the gateway.");
                throw (CompletionException) exception;
            }
            logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
            return response.gatewayId();
        });
}
```

- Untuk detail API, lihat [CreateGateway](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  CreateGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create a Gateway.
 * @param {{ }}
 */
export const main = async ({ gatewayName }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateGatewayCommand({
        gatewayName: gatewayName, // The name to give the created Gateway.
      }),
    );
    console.log("Gateway created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the Gateway.`
      );
    } else {
      throw caught;
    }
  }
};
```


- Untuk detail API, lihat [CreateGateway](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_gateway(self, gateway_name: str, my_thing: str) -> str:
        """
```

Creates an AWS IoT SiteWise Gateway.

```

:param gateway_name: The name of the gateway to create.
:param my_thing: The core device thing name.
:return: The ID of the created gateway.
"""
try:
    response = self.iotsitewise_client.create_gateway(
        gatewayName=gateway_name,
        gatewayPlatform={
            "greengrassV2": {"coreDeviceThingName": my_thing},
        },
        tags={"Environment": "Production"},
    )
    gateway_id = response["gatewayId"]
    return gateway_id
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        logger.error("Gateway %s already exists.", gateway_name)
    else:
        logger.error(
            "Error creating gateway %s. Here's why %s",
            gateway_name,
            err.response["Error"]["Message"],
        )
    raise

```

- Untuk detail API, lihat [CreateGateway](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreatePortal** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreatePortal`.

CLI

AWS CLI

Untuk membuat portal

create-portal Contoh berikut membuat portal web untuk perusahaan peternakan angin. Anda dapat membuat portal hanya di Wilayah yang sama tempat Anda mengaktifkan AWS Single Sign-On.

```
aws iotsitewise create-portal \
  --portal-name WindFarmPortal \
  --portal-description "A portal that contains wind farm projects for Example Corp." \
  --portal-contact-email support@example.com \
  --role-arn arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole
```

Output:

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalStatus": {
    "state": "CREATING"
  },
  "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"
}
```

Untuk informasi selengkapnya, lihat [Memulai SiteWise Monitor AWS IoT](#) di Panduan SiteWise Pengguna AWS IoT dan [Mengaktifkan SSO AWS di](#) Panduan Pengguna IoT..AWS SiteWise

- Untuk detail API, lihat [CreatePortal](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Creates a new IoT SiteWise portal.
 *
 * @param portalName the name of the portal to create.
 * @param iamRole the IAM role ARN to use for the portal.
 * @param contactEmail the email address of the portal contact.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal ID. The calling code
 * can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 * {@link CompletableFuture#get()}.
 *
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 * it available to the calling code as a {@link CompletionException}.
By calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createPortalAsync(String portalName, String
iamRole, String contactEmail) {
    CreatePortalRequest createPortalRequest = CreatePortalRequest.builder()
        .portalName(portalName)
        .portalDescription("This is my custom IoT SiteWise portal.")
        .portalContactEmail(contactEmail)
        .roleArn(iamRole)
        .build();

    return getAsyncClient().createPortal(createPortalRequest)
        .handle((response, exception) -> {
            if (exception != null) {
```

```

        logger.error("Failed to create portal: {} ",
exception.getCause().getMessage());
        throw (CompletionException) exception;
    }
    return response.portalId();
});
}

```

- Untuk detail API, lihat [CreatePortal](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    CreatePortalCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create a Portal.
 * @param {{ portalName: string, portalContactEmail: string, roleArn: string }}
 */
export const main = async ({ portalName, portalContactEmail, roleArn }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const result = await client.send(
            new CreatePortalCommand({
                portalName: portalName, // The name to give the created Portal.
                portalContactEmail: portalContactEmail, // A valid contact email.
                roleArn: roleArn, // The ARN of a service role that allows the portal's
                users to access the portal's resources.
            }),
        );
    }
};

```

```

    console.log("Portal created successfully.");
    return result;
} catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
        console.warn(
            `${caught.message}. There was a problem creating the Portal.`
        );
    } else {
        throw caught;
    }
}
};

```

- Untuk detail API, lihat [CreatePortal](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod

```

```

def from_client(cls) -> "IoTSitewiseWrapper":
    """
    Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
    client.

    :return: An instance of IoTSitewiseWrapper initialized with the default
    AWS IoT SiteWise client.
    """
    iotsitewise_client = boto3.client("iotsitewise")
    return cls(iotsitewise_client)

def create_portal(
    self, portal_name: str, iam_role_arn: str, portal_contact_email: str
) -> str:
    """
    Creates an AWS IoT SiteWise Portal.

    :param portal_name: The name of the portal to create.
    :param iam_role_arn: The ARN of an IAM role.
    :param portal_contact_email: The contact email of the portal.
    :return: The ID of the created portal.
    """
    try:
        response = self.iotsitewise_client.create_portal(
            portalName=portal_name,
            roleArn=iam_role_arn,
            portalContactEmail=portal_contact_email,
        )
        portal_id = response["portalId"]
        waiter = self.iotsitewise_client.get_waiter("portal_active")
        waiter.wait(portalId=portal_id, WaiterConfig={"MaxAttempts": 40})
        return portal_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Portal %s already exists.", portal_name)
        else:
            logger.error(
                "Error creating portal %s. Here's why %s",
                portal_name,
                err.response["Error"]["Message"],
            )
        raise

```

- Untuk detail API, lihat [CreatePortal](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteAsset** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteAsset`.

CLI

AWS CLI

Untuk menghapus aset

`delete-asset` Contoh berikut menghapus aset turbin angin.

```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Output:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

Untuk informasi selengkapnya, lihat [Menghapus aset](#) di Panduan AWS Pengguna SiteWise IoT.

- Untuk detail API, lihat [DeleteAsset](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Deletes an asset.
 *
 * @param assetId the ID of the asset to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
assetId) {
    DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
        .assetId(assetId)
        .build();

    return getAsyncClient().deleteAsset(deleteAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred deleting asset with id: {}",
assetId);
            }
        });
}
```

- Untuk detail API, lihat [DeleteAsset](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  DeleteAssetCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset.
 * @param {{ assetId : string }}
 */
export const main = async ({ assetId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetCommand({
        assetId: assetId, // The model id to delete.
      }),
    );
    console.log("Asset deleted successfully.");
    return { assetDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Untuk detail API, lihat [DeleteAsset](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```
def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise
```

- Untuk detail API, lihat [DeleteAsset](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteAssetModel** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteAssetModel`.

CLI

AWS CLI

Untuk menghapus model aset

`delete-asset-model` Contoh berikut menghapus model aset turbin angin.

```
aws iotsitewise delete-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Output:

```
{
```

```

    "assetModelStatus": {
        "state": "DELETING"
    }
}

```

Untuk informasi selengkapnya, lihat [Menghapus model aset](#) di Panduan AWS Pengguna SiteWise IoT.

- Untuk detail API, lihat [DeleteAssetModel](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh selengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

/**
 * Deletes an Asset Model with the specified ID.
 *
 * @param assetModelId the ID of the Asset Model to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
 *       can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *       {@link CompletableFuture#get()}.
 *       <p>
 *       If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *       it available to the calling code as a {@link CompletionException}.
By calling
 *       {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
    DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
        .assetModelId(assetModelId)

```

```

        .build();

        return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete asset model with ID:{}",
exception.getMessage());
                }
            });
    }
}

```

- Untuk detail API, lihat [DeleteAssetModel](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    DeleteAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset model.
 * @param {{ assetModelId : string }}
 */
export const main = async ({ assetModelId }) => {
    const client = new IoTSiteWiseClient({});
    try {
        await client.send(
            new DeleteAssetModelCommand({
                assetModelId: assetModelId, // The model id to delete.
            }),
        );
    }
};

```

```

    console.log("Asset model deleted successfully.");
    return { assetModelDeleted: true };
} catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
        console.warn(
            `${caught.message}. There was a problem deleting the asset model.`
        );
    } else {
        throw caught;
    }
}
};

```

- Untuk detail API, lihat [DeleteAssetModel](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod

```

```

def from_client(cls) -> "IoTSitewiseWrapper":
    """
    Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
    client.

    :return: An instance of IoTSitewiseWrapper initialized with the default
    AWS IoT SiteWise client.
    """
    iotsitewise_client = boto3.client("iotsitewise")
    return cls(iotsitewise_client)

def delete_asset_model(self, asset_model_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset Model.

    :param asset_model_id: The ID of the asset model to delete.
    """
    try:
        self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset model %s. Here's why %s",
            asset_model_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Untuk detail API, lihat [DeleteAssetModel](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteGateway`.

CLI

AWS CLI

Untuk menghapus gateway

`delete-gateway` Contoh berikut menghapus gateway.

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Menyerap data menggunakan gateway](#) di Panduan Pengguna AWS SiteWise IoT.

- Untuk detail API, lihat [DeleteGateway](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**  
 * Deletes the specified gateway.  
 *  
 * @param gatewayId the ID of the gateway to delete.  
 * @return a {@link CompletableFuture} that represents a {@link  
DeleteGatewayResponse} result.. The calling code  
 *       can attach callbacks, then handle the result or exception by  
calling {@link CompletableFuture#join()} or  
 *       {@link CompletableFuture#get()}.  
 *       <p>  
 *       If any completion stage in this method throws an exception, the  
method logs the exception cause and keeps  
 *       it available to the calling code as a {@link CompletionException}.  
By calling
```

```

    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
        DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
            .gatewayId(gatewayId)
            .build();

        return getAsyncClient().deleteGateway(deleteGatewayRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
                }
            });
    }
}

```

- Untuk detail API, lihat [DeleteGateway](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    DeleteGatewayCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */

```

```
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway deleted successfully.");
    return { gatewayDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Untuk detail API, lihat [DeleteGateway](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.
        """
```

```

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.

        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def delete_gateway(self, gateway_id: str) -> None:
        """
        Deletes an AWS IoT SiteWise Gateway.

        :param gateway_id: The ID of the gateway to delete.
        """
        try:
            self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error("Gateway %s does not exist.", gateway_id)
            else:
                logger.error(
                    "Error deleting gateway %s. Here's why %s",
                    gateway_id,
                    err.response["Error"]["Message"],
                )
            raise

```

- Untuk detail API, lihat [DeleteGateway](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeletePortal** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeletePortal`.

CLI

AWS CLI

Untuk menghapus portal

`delete-portal` Contoh berikut menghapus portal web untuk perusahaan peternakan angin.

```
aws iotsitewise delete-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE
```

Output:

```
{
  "portalStatus": {
    "state": "DELETING"
  }
}
```

Untuk informasi selengkapnya, lihat [Menghapus portal](#) di Panduan Pengguna AWS SiteWise IoT.

- Untuk detail API, lihat [DeletePortal](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh selengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

/**
 * Deletes a portal.
 *
 * @param portalId the ID of the portal to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeletePortalResponse}. The calling code can attach
 *         callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeletePortalResponse> deletePortalAsync(String
portalId) {
    DeletePortalRequest deletePortalRequest = DeletePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().deletePortal(deletePortalRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete portal with ID: {}. Error:
{}", portalId, exception.getCause().getMessage());
            }
        });
}

```

- Untuk detail API, lihat [DeletePortal](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  DeletePortalCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * List asset models.
 * @param {{ portalId : string }}
 */
export const main = async ({ portalId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeletePortalCommand({
        portalId: portalId, // The id of the portal.
      }),
    );
    console.log("Portal deleted successfully.");
    return { portalDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the portal. Please check the portal id.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Untuk detail API, lihat [DeletePortal](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def delete_portal(self, portal_id: str) -> None:
        """
```



```

Deletes an AWS IoT SiteWise Portal.

:param portal_id: The ID of the portal to delete.
"""
try:
    self.iotsitewise_client.delete_portal(portalId=portal_id)
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Portal %s does not exist.", portal_id)
    else:
        logger.error(
            "Error deleting portal %s. Here's why %s",
            portal_id,
            err.response["Error"]["Message"],
        )
    raise

```

- Untuk detail API, lihat [DeletePortal](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeAssetModel** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeAssetModel`.

CLI

AWS CLI

Untuk menggambarkan model aset

`describe-asset-model` Contoh berikut menjelaskan model aset ladang angin.

```

aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-2222EXAMPLE

```

Output:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind turbines",
  "assetModelProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
              "name": "power",
              "value": {
                "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE",
                "hierarchyId": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE"
              }
            }
          ]
        },
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": " "
        }
      }
    }
  ]
}

```

```

    }
  }
],
"assetModelHierarchies": [
  {
    "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
    "name": "Wind Turbines",
    "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
],
"assetModelCreationDate": 1575671284.0,
"assetModelLastUpdateDate": 1575671988.0,
"assetModelStatus": {
  "state": "ACTIVE"
}
}

```

Untuk informasi selengkapnya, lihat [Menjelaskan model aset tertentu](#) di Panduan Pengguna AWS SiteWise IoT.

- Untuk detail API, lihat [DescribeAssetModel](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

/**
 * Retrieves the property IDs associated with a specific asset model.
 *
 * @param assetModelId the ID of the asset model that defines the properties.
 * @return a {@link CompletableFuture} that represents a {@link Map} result
 * that associates the property name to the
 *         property ID. The calling code can attach callbacks, then handle the
 * result or exception by calling
 *         {@link CompletableFuture#join()} or {@link
 * CompletableFuture#get()}.

```

```

    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
        ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
        return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
            .handle((response, throwable) -> {
                if (response != null) {
                    return response.assetModelPropertySummaries().stream()
                        .collect(Collectors
                            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
                } else {
                    logger.error("Error occurred while fetching property IDs:
{}.\"", throwable.getCause().getMessage());
                    throw (CompletionException) throwable;
                }
            });
    }
}

```

- Untuk detail API, lihat [DescribeAssetModel](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    DescribeAssetModelCommand,

```

```

    IoTSiteWiseClient,
  } from "@aws-sdk/client-iotsitewise";
  import { parseArgs } from "node:util";

  /**
   * Describe an asset model.
   * @param {{ assetModelId : string }}
   */
  export const main = async ({ assetModelId }) => {
    const client = new IoTSiteWiseClient({});
    try {
      const { assetModelDescription } = await client.send(
        new DescribeAssetModelCommand({
          assetModelId: assetModelId, // The ID of the Gateway to describe.
        }),
      );
      console.log("Asset model information retrieved successfully.");
      return { assetModelDescription: assetModelDescription };
    } catch (caught) {
      if (caught instanceof Error && caught.name === "ResourceNotFound") {
        console.warn(
          `${caught.message}. The asset model could not be found. Please check the asset model id.`
        );
      } else {
        throw caught;
      }
    }
  };

```

- Untuk detail API, lihat [DescribeAssetModel](#) di Referensi AWS SDK untuk JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeGateway`.

CLI

AWS CLI

Untuk menggambarkan gateway

`describe-gateway` Contoh berikut menjelaskan gateway.

```
aws iotsitewise describe-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Output:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayName": "ExampleCorpGateway",
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayPlatform": {
    "greengrass": {
      "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"
    }
  },
  "gatewayCapabilitySummaries": [
    {
      "capabilityNamespace": "iotsitewise:opcuacollector:1",
      "capabilitySyncStatus": "IN_SYNC"
    }
  ],
  "creationDate": 1588369971.457,
  "lastUpdateDate": 1588369971.457
}
```

Untuk informasi selengkapnya, lihat [Menyerap data menggunakan gateway](#) di Panduan Pengguna AWS SiteWise IoT.

- Untuk detail API, lihat [DescribeGateway](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Describes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to describe.
 * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
    DescribeGatewayRequest request = DescribeGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().describeGateway(request)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
            }
        });
}
```

- Untuk detail API, lihat [DescribeGateway](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  DescribeGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const { gatewayDescription } = await client.send(
      new DescribeGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway information retrieved successfully.");
    return { gatewayDescription: gatewayDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`,
      );
    } else {
      throw caught;
    }
  }
}
```



```
};
```

- Untuk detail API, lihat [DescribeGateway](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```
def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise
```

- Untuk detail API, lihat [DescribeGateway](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribePortal** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribePortal`.

CLI

AWS CLI

Untuk menggambarkan portal

`describe-portal` Contoh berikut menjelaskan portal web untuk perusahaan peternakan angin.

```
aws iotsitewise describe-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Output:

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  },
  "portalCreationDate": "2020-02-04T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",
  "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"
}
```

Untuk informasi selengkapnya, lihat [Mengelola portal Anda](#) di Panduan Pengguna AWS SiteWise IoT.

- Untuk detail API, lihat [DescribePortal](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
```

```

    * Retrieves a portal's description.
    *
    * @param portalId the ID of the portal to describe.
    * @return a {@link CompletableFuture} that represents a {@link String}
    result of the portal's start URL
    *      (see: {@link DescribePortalResponse#portalStartUrl()}). The
    calling code can attach callbacks, then handle the
    *      result or exception by calling {@link CompletableFuture#join()} or
    {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
    method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
    By calling
    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<String> describePortalAsync(String portalId) {
        DescribePortalRequest request = DescribePortalRequest.builder()
            .portalId(portalId)
            .build();

        return getAsyncClient().describePortal(request)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("An exception occurred retrieving the portal
description: {}", exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.portalStartUrl();
            });
    }
}

```

- Untuk detail API, lihat [DescribePortal](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  DescribePortalCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe a portal.
 * @param {{ portalId: string }}
 */
export const main = async ({ portalId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new DescribePortalCommand({
        portalId: portalId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Portal information retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Portal could not be found. Please check the Portal Id.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Untuk detail API, lihat [DescribePortal](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_gateway(self, gateway_name: str, my_thing: str) -> str:
        """
```

Creates an AWS IoT SiteWise Gateway.

```
:param gateway_name: The name of the gateway to create.
:param my_thing: The core device thing name.
:return: The ID of the created gateway.
"""
try:
    response = self.iotsitewise_client.create_gateway(
        gatewayName=gateway_name,
        gatewayPlatform={
            "greengrassV2": {"coreDeviceThingName": my_thing},
        },
        tags={"Environment": "Production"},
    )
    gateway_id = response["gatewayId"]
    return gateway_id
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        logger.error("Gateway %s already exists.", gateway_name)
    else:
        logger.error(
            "Error creating gateway %s. Here's why %s",
            gateway_name,
            err.response["Error"]["Message"],
        )
    raise
```

- Untuk detail API, lihat [DescribePortal](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetAssetPropertyValue** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetAssetPropertyValue`.

CLI

AWS CLI

Untuk mengambil nilai aset properti saat ini

`get-asset-property-value` Contoh berikut mengambil daya total aset turbin angin saat ini.

```
aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE
```

Output:

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

Untuk informasi selengkapnya, lihat [Menanyakan nilai properti aset saat ini](#) di Panduan AWS Pengguna SiteWise IoT.

- Untuk detail API, lihat [GetAssetPropertyValue](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).


```

/**
 * Fetches the value of an asset property.
 *
 * @param propId the ID of the asset property to fetch.
 * @param assetId the ID of the asset to fetch the property value for.
 * @return a {@link CompletableFuture} that represents a {@link Double}
result. The calling code can attach
 *      callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
    GetAssetPropertyValueRequest assetPropertyValueRequest =
GetAssetPropertyValueRequest.builder()
        .propertyId(propId)
        .assetId(assetId)
        .build();

    return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.propertyValue().value().doubleValue();
        });
}

```

- Untuk detail API, lihat [GetAssetPropertyValue](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import {
  GetAssetPropertyValueCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe an asset property value.
 * @param {{ entryId : string }}
 */
export const main = async ({ entryId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new GetAssetPropertyValueCommand({
        entryId: entryId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Asset property information retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The asset property entry could not be found. Please check the entry id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Untuk detail API, lihat [GetAssetPropertyValue](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def get_asset_property_value(
        self, asset_id: str, property_id: str
```

```

) -> Dict[str, Any]:
    """
    Gets the value of an AWS IoT SiteWise Asset Property.

    :param asset_id: The ID of the asset.
    :param property_id: The ID of the property.
    :return: A dictionary containing the value of the property.
    """
    try:
        response = self.iotsitewise_client.get_asset_property_value(
            assetId=asset_id, propertyId=property_id
        )
        return response["propertyValue"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error(
                "Asset %s or property %s does not exist.", asset_id,
property_id
            )
        else:
            logger.error(
                "Error getting asset property value. Here's why %s",
                err.response["Error"]["Message"],
            )
        raise

```

- Untuk detail API, lihat [GetAssetPropertyValue](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ListAssetModels** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ListAssetModels`.

CLI

AWS CLI

Untuk membuat daftar semua model aset

`list-asset-models` Contoh berikut mencantumkan semua model aset yang ditentukan di AWS akun Anda di Wilayah saat ini.

```
aws iotsitewise list-asset-models
```

Output:

```
{
  "assetModelSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "Wind Farm Model",
      "description": "Represents a wind farm that comprises many wind turbines",
      "creationDate": 1575671284.0,
      "lastUpdateDate": 1575671988.0,
      "status": {
        "state": "ACTIVE"
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "name": "Wind Turbine Model",
      "description": "Represents a wind turbine manufactured by Example Corp",
      "creationDate": 1575671207.0,
      "lastUpdateDate": 1575686273.0,
      "status": {
        "state": "ACTIVE"
      }
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Mencantumkan semua model aset](#) di SiteWise Panduan Pengguna AWS IoT.

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh selengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * Retrieves the asset model ID for the given asset model name.
 *
 * @param assetModelName the name of the asset model for the ID.
 * @return a {@link CompletableFuture} that represents a {@link String}
 * result of the asset model ID or null if the
 *      asset model cannot be found. The calling code can attach
 * callbacks, then handle the result or exception
 *      by calling {@link CompletableFuture#join()} or {@link
 * CompletableFuture#get()}.
 *
 *      <p>
 *      If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
 * By calling
 *      {@link CompletionException#getCause()}, the calling code can
 * access the original exception.
 */
public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
{
    ListAssetModelsRequest listAssetModelsRequest =
ListAssetModelsRequest.builder().build();
    return getAsyncClient().listAssetModels(listAssetModelsRequest)
        .handle((listAssetModelsResponse, exception) -> {
            if (exception != null) {
                logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
            }
        });
}
```

```

        throw (CompletionException) exception;
    }
    for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {
        if (assetModelSummary.name().equals(assetModelName)) {
            return assetModelSummary.id();
        }
    }
    return null;
});
}

```

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import {
    ListAssetModelsCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * List asset models.
 * @param {{ assetModelTypes : array }}
 */
export const main = async ({ assetModelTypes = [] }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const result = await client.send(
            new ListAssetModelsCommand({
                assetModelTypes: assetModelTypes, // The model types to list
            }),

```

```

    );
    console.log("Asset model types retrieved successfully.");
    return result;
} catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
        console.warn(
            `${caught.message}. There was a problem listing the asset model types.`
        );
    } else {
        throw caught;
    }
}
};

```

- Untuk detail API, lihat [ListAssetModels](#) di Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

```



```

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def list_asset_models(self) -> List[Dict[str, Any]]:
        """
        Lists all AWS IoT SiteWise Asset Models.

        :return: A list of dictionaries containing information about each asset
        model.

        """
        try:
            asset_models = []
            paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
            pages = paginator.paginate()
            for page in pages:
                asset_models.extend(page["assetModelSummaries"])
            return asset_models
        except ClientError as err:
            logger.error(
                "Error listing asset models. Here's why %s",
                err.response["Error"]["Message"],
            )
            raise

```

- Untuk detail API, lihat [ListAssetModels](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di AWS IoT SiteWise

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [program AWS kepatuhan program AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku AWS IoT SiteWise, lihat [AWS layanan dalam cakupan berdasarkan AWS layanan program kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS IoT SiteWise. Topik berikut menunjukkan cara mengonfigurasi AWS IoT SiteWise untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS IoT SiteWise sumber daya Anda.

Topik

- [Perlindungan data di AWS IoT SiteWise](#)
- [Enkripsi data di AWS IoT SiteWise](#)
- [Identitas dan manajemen akses untuk AWS IoT SiteWise](#)
- [Validasi kepatuhan untuk AWS IoT SiteWise](#)
- [Ketahanan di AWS IoT SiteWise](#)
- [Keamanan infrastruktur di AWS IoT SiteWise](#)
- [Analisis konfigurasi dan kerentanan di AWS IoT SiteWise](#)
- [Titik akhir VPC untuk AWS IoT SiteWise](#)
- [Praktik terbaik keamanan untuk AWS IoT SiteWise](#)

Perlindungan data di AWS IoT SiteWise

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS IoT SiteWise. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS IoT SiteWise atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

- [Privasi lalu lintas internetwork untuk AWS IoT SiteWise](#)
- [AWS IoT SiteWise Peningkatan Layanan Asisten Bisnis](#)

Privasi lalu lintas internetwork untuk AWS IoT SiteWise

Koneksi antara AWS IoT SiteWise dan aplikasi lokal, seperti gateway SiteWise Edge, diamankan melalui koneksi Transport Layer Security (TLS). Untuk informasi selengkapnya, lihat [Enkripsi data dalam perjalanan untuk AWS IoT SiteWise](#).

AWS IoT SiteWise tidak mendukung koneksi antara Availability Zone dalam AWS Wilayah atau koneksi antar AWS akun.

Anda dapat mengonfigurasi Pusat Identitas IAM hanya dalam satu Wilayah pada satu waktu. SiteWise Monitor terhubung ke Wilayah yang Anda konfigurasi untuk IAM Identity Center. Ini berarti Anda menggunakan satu Wilayah untuk akses Pusat Identitas IAM, tetapi Anda dapat membuat portal di Wilayah mana pun.

AWS IoT SiteWise Peningkatan Layanan Asisten Bisnis

AWS IoT SiteWise Asisten tidak menggunakan data pelanggan untuk peningkatan layanan atau untuk meningkatkan yang mendasarinya LLMs.

Enkripsi data di AWS IoT SiteWise

Enkripsi data mengacu pada perlindungan data saat dalam perjalanan (saat bepergian ke dan dari AWS IoT SiteWise, dan antara gateway dan server SiteWise Edge), dan saat istirahat (saat disimpan di perangkat lokal atau di AWS layanan). Anda dapat melindungi data saat transit menggunakan Keamanan Lapisan Pengangkutan (TLS) atau saat istirahat menggunakan enkripsi di sisi klien.

Note

AWS IoT SiteWise mengekspos pemrosesan tepi APIs yang di-host dalam gateway SiteWise Edge dan dapat diakses melalui jaringan lokal. Ini APIs diekspos melalui koneksi TLS yang

didukung oleh server-sertifikat yang dimiliki oleh konektor Edge. AWS IoT SiteWise Untuk otentikasi klien, ini APIs menggunakan kata sandi kontrol akses. Server-certificate private-key dan password akses-kontrol keduanya disimpan pada disk. AWS IoT SiteWise pemrosesan tepi bergantung pada enkripsi sistem file untuk keamanan kredensial ini saat istirahat.

Untuk informasi lebih lanjut tentang enkripsi sisi server dan enkripsi sisi klien, tinjau topik yang tercantum di bawah ini.

Topik

- [Enkripsi saat istirahat AWS IoT SiteWise](#)
- [Enkripsi data dalam perjalanan untuk AWS IoT SiteWise](#)
- [Manajemen kunci di AWS IoT SiteWise](#)

Enkripsi saat istirahat AWS IoT SiteWise

AWS IoT SiteWise menyimpan data Anda di AWS Cloud dan di gateway AWS IoT SiteWise Edge.

Data saat istirahat di AWS Cloud

AWS IoT SiteWise menyimpan data di AWS layanan lain yang mengenkripsi data saat istirahat secara default. Enkripsi saat istirahat terintegrasi dengan AWS Key Management Service (AWS KMS) untuk mengelola kunci enkripsi yang digunakan untuk mengenkripsi nilai properti aset Anda dan nilai agregat di. AWS IoT SiteWise Anda dapat memilih untuk menggunakan kunci terkelola pelanggan untuk mengenkripsi nilai properti aset dan nilai agregat. AWS IoT SiteWise Anda dapat membuat, mengelola, dan melihat kunci enkripsi Anda melalui AWS KMS.

Anda dapat memilih Kunci milik AWS untuk mengenkripsi data Anda, atau memilih kunci yang dikelola pelanggan untuk mengenkripsi nilai properti aset dan nilai agregat Anda:

Cara kerjanya

Enkripsi saat istirahat terintegrasi dengan AWS KMS untuk mengelola kunci enkripsi yang digunakan untuk mengenkripsi data Anda.

- Kunci milik AWS — Kunci enkripsi default. AWS IoT SiteWise memiliki kunci ini. Anda tidak dapat melihat kunci ini di AWS akun Anda. Anda juga tidak dapat melihat operasi pada kunci di AWS CloudTrail log. Anda dapat menggunakan kunci ini tanpa biaya tambahan.

- Kunci yang dikelola pelanggan — Kunci disimpan di akun Anda, yang Anda buat, miliki, dan kelola. Anda memiliki kontrol penuh atas tombol KMS. AWS KMS Biaya tambahan berlaku.

Kunci milik AWS

Kunci milik AWS tidak disimpan di akun Anda. Mereka adalah bagian dari kumpulan kunci KMS yang AWS memiliki dan mengelola untuk digunakan di beberapa AWS akun. AWS Layanan dapat digunakan Kunci milik AWS untuk melindungi data Anda.

Anda tidak dapat melihat, mengelola, menggunakan Kunci milik AWS, atau mengaudit penggunaannya. Namun, Anda tidak perlu melakukan pekerjaan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda.

Anda tidak dikenakan biaya bulanan atau biaya penggunaan jika Anda menggunakan Kunci milik AWS, dan mereka tidak dihitung terhadap AWS KMS kuota untuk akun Anda.

Kunci yang dikelola pelanggan

Kunci yang dikelola pelanggan adalah kunci KMS di akun Anda yang Anda buat, miliki, dan kelola. Anda memiliki kontrol penuh atas kunci KMS ini, seperti berikut ini:

- Menetapkan dan memelihara kebijakan utama mereka, kebijakan IAM, dan hibah
- Mengaktifkan dan menonaktifkannya
- Memutar materi kriptografi mereka
- Menambahkan tanda
- Membuat alias yang merujuk kepada mereka
- Menjadwalkan mereka untuk dihapus

Anda juga dapat menggunakan CloudTrail dan Amazon CloudWatch Logs untuk melacak permintaan yang AWS IoT SiteWise dikirimkan AWS KMS atas nama Anda.

Jika Anda menggunakan kunci yang dikelola pelanggan, Anda harus memberikan AWS IoT SiteWise akses ke kunci KMS yang disimpan di akun Anda. AWS IoT SiteWise menggunakan amplop dan hierarki kunci untuk mengenkripsi data. Kunci AWS KMS enkripsi Anda digunakan untuk mengenkripsi kunci root hierarki kunci ini. Untuk informasi lebih lanjut, lihat [Enkripsi amplop](#) di Panduan Developer AWS Key Management Service .

Contoh kebijakan berikut memberikan AWS IoT SiteWise izin untuk membuat kunci terkelola pelanggan atas nama Anda. Ketika Anda membuat kunci Anda, Anda harus mengizinkan `kms:CreateGrant` dan `kms:DescribeKey` tindakan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Konteks enkripsi untuk hibah yang Anda buat menggunakan ID Anda `aws:iotsitewise:subscriberId` dan akun.

Data diam di gateway SiteWise Edge

AWS IoT SiteWise gateway menyimpan data berikut pada sistem file lokal:

- Informasi konfigurasi sumber OPC UA
- Kumpulan jalur aliran data OPC UA dari sumber OPC UA yang terhubung
- Data industri di-cache saat gateway SiteWise Edge kehilangan koneksi ke internet

SiteWise Gerbang tepi berjalan. AWS IoT Greengrass AWS IoT Greengrass bergantung pada izin file Unix dan enkripsi full-disk (jika diaktifkan) untuk melindungi data pada intinya. Anda bertanggung jawab untuk mengamankan sistem file dan perangkat.

Namun, AWS IoT Greengrass mengenkripsi salinan lokal rahasia server OPC UA Anda yang diambil dari Secrets Manager. Untuk informasi selengkapnya, lihat [Enkripsi rahasia](#) di Panduan AWS IoT Greengrass Version 1 Pengembang.

Untuk informasi selengkapnya tentang enkripsi saat istirahat pada AWS IoT Greengrass inti, lihat [Enkripsi saat istirahat](#) di Panduan AWS IoT Greengrass Version 1 Pengembang.

Enkripsi data dalam perjalanan untuk AWS IoT SiteWise

AWS IoT SiteWise menggunakan enkripsi dalam perjalanan untuk mengamankan data yang dikirimkan antara perangkat, gateway, dan Cloud Anda. AWS Komunikasi dengan AWS IoT SiteWise dienkripsi menggunakan HTTPS dan TLS 1.2, memastikan bahwa data Anda tetap rahasia dan terlindungi dari akses atau intersepsi yang tidak sah.

Ada tiga mode komunikasi di mana data dalam perjalanan:

- [Melalui internet](#) — Komunikasi antara perangkat lokal (termasuk gateway SiteWise Edge) dan AWS IoT SiteWise dienkripsi.
- [Melalui jaringan lokal](#) — Komunikasi antara OpsHub untuk SiteWise aplikasi dan gateway SiteWise Edge selalu dienkripsi. Komunikasi antara aplikasi SiteWise monitor yang berjalan di dalam browser Anda dan gateway SiteWise Edge selalu dienkripsi. Komunikasi antara gateway SiteWise Edge dan sumber OPC UA dapat dienkripsi.
- [Antara komponen di gateway SiteWise Edge](#) — Komunikasi antar AWS IoT Greengrass komponen di gateway SiteWise Edge tidak dienkripsi.

Topik

- [Data dalam transit melalui internet](#)
- [Data dalam transit melalui jaringan lokal](#)
- [Data dalam perjalanan antar komponen lokal di SiteWise Edge](#)

Data dalam transit melalui internet

AWS IoT SiteWise Menggunakan Transport Layer Security (TLS) untuk mengenkripsi semua komunikasi melalui internet. Semua data yang dikirim ke AWS Cloud dikirim melalui koneksi TLS menggunakan protokol MQTT atau HTTPS, sehingga aman secara default. SiteWise Edge gateway, yang berjalan pada AWS IoT Greengrass, dan pemberitahuan nilai properti menggunakan model keamanan AWS IoT transportasi. Untuk informasi lebih lanjut, lihat [Keamanan angkutan](#) di dalam AWS IoT Panduan Developer.

Data dalam transit melalui jaringan lokal

SiteWise Edge gateway mengikuti spesifikasi OPC UA untuk komunikasi dengan sumber OPC UA lokal. Anda bertanggung jawab untuk mengonfigurasi sumber Anda agar menggunakan mode keamanan pesan yang mengenkripsi data dalam perjalanan.

Jika Anda memilih mode keamanan pesan tanda tangan, data yang sedang transit antara gateway SiteWise Edge dan sumber ditandatangani tetapi tidak dienkripsi. Jika Anda memilih mode keamanan pesan tanda dan enkripsi, data yang sedang transit antara gateway dan sumber SiteWise Edge ditandatangani dan dienkripsi. Untuk informasi selengkapnya tentang mengonfigurasi sumber, lihat [Tambahkan sumber data ke gateway AWS IoT SiteWise Edge Anda](#).

Komunikasi antara aplikasi konsol tepi dan gateway SiteWise Edge selalu dienkripsi oleh TLS. Konektor SiteWise SiteWise Edge pada gateway Edge menghasilkan dan menyimpan sertifikat yang ditandatangani sendiri untuk dapat membuat koneksi TLS dengan konsol tepi untuk AWS IoT SiteWise aplikasi. Anda harus menyalin sertifikat ini dari gateway SiteWise Edge Anda ke konsol tepi untuk AWS IoT SiteWise aplikasi sebelum Anda menghubungkan aplikasi ke gateway SiteWise Edge. Ini memastikan bahwa konsol tepi untuk AWS IoT SiteWise aplikasi dapat memverifikasi bahwa ia telah terhubung ke gateway SiteWise Edge tepercaya Anda.

Selain TLS untuk kerahasiaan dan keaslian server, SiteWise Edge menggunakan protokol SiGv4 untuk menetapkan keaslian aplikasi konsol tepi. Konektor SiteWise Edge pada gateway SiteWise Edge menerima dan menyimpan kata sandi untuk dapat memverifikasi koneksi masuk dari aplikasi konsol tepi, aplikasi SiteWise Monitor yang berjalan di dalam browser, dan klien lain berdasarkan SDK. AWS IoT SiteWise

Untuk informasi selengkapnya tentang membuat kata sandi dan sertifikat server, lihat [the section called “Kelola gateway”](#).

Data dalam perjalanan antar komponen lokal di SiteWise Edge

SiteWise Edge gateway berjalan AWS IoT Greengrass, yang tidak mengenkripsi data yang dipertukarkan secara lokal pada AWS IoT Greengrass inti karena data tidak meninggalkan perangkat. Ini termasuk komunikasi antar AWS IoT Greengrass komponen seperti AWS IoT SiteWise konektor. Untuk informasi selengkapnya, lihat [Data pada perangkat inti](#) di Panduan AWS IoT Greengrass Version 1 Pengembang.

Manajemen kunci di AWS IoT SiteWise

AWS IoT SiteWise manajemen kunci cloud

Secara default, AWS IoT SiteWise gunakan Kunci yang dikelola AWS untuk melindungi data Anda di AWS Cloud. Anda dapat memperbarui pengaturan untuk menggunakan kunci yang dikelola pelanggan untuk mengenkripsi beberapa data. AWS IoT SiteWise Anda dapat membuat, mengelola, dan melihat kunci enkripsi Anda melalui AWS Key Management Service (AWS KMS).

AWS IoT SiteWise mendukung enkripsi sisi server dengan kunci terkelola pelanggan yang disimpan AWS KMS untuk mengenkripsi data berikut:

- Nilai properti aset
- Nilai agregat

 Note

Data dan sumber daya lainnya dienkripsi menggunakan enkripsi default dengan kunci yang dikelola oleh AWS IoT SiteWise Kunci ini disimpan di AWS IoT SiteWise akun.

Untuk informasi lebih lanjut, lihat [Apa itu AWS Key Management Service?](#) di Panduan AWS Key Management Service Pengembang.

Aktifkan enkripsi menggunakan kunci yang dikelola pelanggan

Untuk menggunakan kunci yang dikelola pelanggan AWS IoT SiteWise, Anda perlu memperbarui AWS IoT SiteWise pengaturan Anda.

Untuk mengaktifkan enkripsi menggunakan kunci KMS

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Pilih Pengaturan Akun dan pilih Edit untuk membuka halaman Edit pengaturan akun.
3. Untuk jenis kunci Enkripsi, pilih Pilih AWS KMS kunci yang berbeda. Ini memungkinkan enkripsi dengan kunci terkelola pelanggan yang disimpan di AWS KMS.

 Note

Saat ini, Anda hanya dapat menggunakan enkripsi kunci terkelola pelanggan untuk nilai properti aset dan nilai agregat.

4. Pilih kunci KMS Anda dengan salah satu opsi berikut:
 - Untuk menggunakan kunci KMS yang ada — Pilih alias kunci KMS Anda dari daftar.
 - Untuk membuat kunci KMS baru — Pilih Buat AWS KMS kunci.

 Note

Ini membuka AWS KMS dasbor. Untuk informasi selengkapnya tentang membuat kunci KMS, lihat [Membuat kunci](#) di Panduan AWS Key Management Service Pengembang.

5. Pilih Simpan untuk memperbarui pengaturan Anda.

SiteWise Manajemen kunci gateway tepi

SiteWise Edge gateway berjalan AWS IoT Greengrass, dan perangkat AWS IoT Greengrass ini menggunakan kunci publik dan pribadi untuk mengautentikasi dengan AWS Cloud dan mengenkripsi rahasia lokal, seperti rahasia otentikasi OPC UA. Untuk informasi selengkapnya, lihat [Manajemen kunci](#) di Panduan AWS IoT Greengrass Version 1 Pengembang.

Identitas dan manajemen akses untuk AWS IoT SiteWise

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS IoT SiteWise IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens untuk AWS IoT SiteWise keamanan](#)
- [Otentikasi dengan identitas di AWS IoT SiteWise](#)
- [Bagaimana AWS IoT SiteWise bekerja dengan IAM](#)
- [AWS kebijakan terkelola untuk AWS IoT SiteWise](#)
- [Gunakan peran terkait layanan untuk AWS IoT SiteWise](#)
- [Siapkan izin untuk alarm acara di AWS IoT SiteWise](#)
- [Cross-service membingungkan deputy pencegahan di AWS IoT SiteWise](#)
- [Memecahkan masalah AWS IoT SiteWise identitas dan akses](#)

Audiens untuk AWS IoT SiteWise keamanan

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS IoT SiteWise

Pengguna layanan — Jika Anda menggunakan AWS IoT SiteWise layanan untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensial dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS IoT SiteWise fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS IoT SiteWise, lihat [Memecahkan masalah AWS IoT SiteWise identitas dan akses](#).

Administrator layanan — Jika Anda bertanggung jawab atas AWS IoT SiteWise sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS IoT SiteWise. Tugas Anda adalah menentukan AWS IoT SiteWise fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS IoT SiteWise, lihat [Bagaimana AWS IoT SiteWise bekerja dengan IAM](#).

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS IoT SiteWise. Untuk melihat contoh kebijakan AWS IoT SiteWise berbasis identitas yang dapat Anda gunakan di IAM, lihat [AWS IoT SiteWise contoh kebijakan berbasis identitas](#)

Otentikasi dengan identitas di AWS IoT SiteWise

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Autentikasi multi-faktor AWS di IAM](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat [Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat [beralih dari pengguna ke peran IAM \(konsol\)](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Buat peran untuk penyedia identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy).

Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses maju](#).
- Peran layanan – Peran layanan adalah [peran IAM](#) yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna](#) IAM.

Bagaimana AWS IoT SiteWise bekerja dengan IAM

Sebelum Anda menggunakan AWS Identity and Access Management (IAM) untuk mengelola akses AWS IoT SiteWise, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan. AWS IoT SiteWise

Fitur IAM	Didukung oleh AWS IoT SiteWise
Kebijakan berbasis identitas dengan izin tingkat sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
Kebijakan berbasis sumber daya	Tidak
Daftar kontrol akses (ACLs)	Tidak
Otorisasi berbasis tag (ABAC)	Ya
Kredensial sementara	Ya
Sesi akses teruskan (FAS)	Ya
Peran terkait layanan	Ya
Peran layanan	Ya

Untuk mendapatkan pandangan tingkat tinggi tentang bagaimana AWS IoT SiteWise dan AWS layanan lain bekerja dengan IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Daftar Isi

- [AWS IoT SiteWise Peran IAM](#)
 - [Gunakan kredensial sementara dengan AWS IoT SiteWise](#)
 - [Sesi akses teruskan \(FAS\) untuk AWS IoT SiteWise](#)
 - [Peran terkait layanan](#)
 - [Peran layanan](#)
 - [Pilih peran IAM di AWS IoT SiteWise](#)
- [Otorisasi berdasarkan tanda AWS IoT SiteWise](#)
- [AWS IoT SiteWise kebijakan berbasis identitas](#)
 - [Tindakan kebijakan](#)
 - [BatchPutAssetPropertyValue otorisasi](#)
 - [Sumber daya kebijakan](#)
 - [Kunci kondisi kebijakan](#)
 - [Contoh](#)
- [AWS IoT SiteWise contoh kebijakan berbasis identitas](#)
 - [Praktik terbaik kebijakan](#)
 - [Gunakan AWS IoT SiteWise konsol](#)
 - [Mengizinkan pengguna melihat izin mereka sendiri](#)
 - [Izinkan pengguna untuk menelan data ke aset dalam satu hierarki](#)
 - [Lihat AWS IoT SiteWise aset berdasarkan tag](#)
- [Mengelola akses menggunakan kebijakan di AWS IoT SiteWise](#)
 - [Kebijakan berbasis identitas](#)
 - [Kebijakan berbasis sumber daya](#)
 - [Daftar kontrol akses \(ACLs\)](#)
 - [Jenis-jenis kebijakan lain](#)
 - [Berbagai jenis kebijakan](#)

AWS IoT SiteWise Peran IAM

[Peran](#) IAM adalah entitas di dalam akun AWS Anda yang memiliki izin tertentu.

Gunakan kredensial sementara dengan AWS IoT SiteWise

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensial keamanan sementara dengan memanggil operasi AWS STS API seperti [AssumeRole](#) atau [GetFederationToken](#).

AWS IoT SiteWise mendukung menggunakan kredensi sementara.

SiteWise Monitor mendukung pengguna federasi untuk mengakses portal. Pengguna portal mengautentikasi dengan Pusat Identitas IAM atau kredensial IAM mereka.

Important

Pengguna atau peran harus memiliki `iotsitewise:DescribePortal` izin untuk masuk ke portal.

Saat pengguna masuk ke portal, SiteWise Monitor akan membuat kebijakan sesi yang memberikan izin berikut:

- Akses hanya-baca ke aset dan data aset AWS IoT SiteWise di akun Anda yang peran portal tersebut menyediakan akses.
- Akses ke proyek di portal tempat pengguna memiliki akses administrator (pemilik proyek) atau hanya-baca (penampil proyek).

Untuk informasi selengkapnya tentang izin pengguna portal federasi, lihat [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#).

Sesi akses teruskan (FAS) untuk AWS IoT SiteWise

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses maju](#).

Peran terkait layanan

[Peran terkait AWS layanan](#) memungkinkan layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran terkait layanan muncul di AWS akun Anda dan dimiliki oleh layanan. Administrator IAM dapat melihat tetapi tidak dapat mengedit izin untuk peran terkait layanan.

AWS IoT SiteWise mendukung peran terkait layanan. Untuk informasi selengkapnya tentang cara membuat atau mengelola peran terkait layanan AWS IoT SiteWise, lihat [Gunakan peran terkait layanan untuk AWS IoT SiteWise](#).

Peran layanan

Fitur ini memungkinkan layanan untuk menerima [peran layanan](#) atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di AWS akun Anda dan dimiliki oleh akun. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

AWS IoT SiteWise menggunakan peran layanan untuk memungkinkan pengguna portal SiteWise Monitor mengakses beberapa sumber AWS IoT SiteWise daya Anda atas nama Anda. Untuk informasi selengkapnya, lihat [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#).

Anda harus memiliki izin yang diperlukan sebelum Anda dapat membuat model AWS IoT Events alarm di AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Siapkan izin untuk alarm acara di AWS IoT SiteWise](#).

Pilih peran IAM di AWS IoT SiteWise

Saat Anda membuat portal sumber daya AWS IoT SiteWise, Anda harus memilih peran untuk memungkinkan pengguna federasi portal SiteWise Monitor Anda mengakses AWS IoT SiteWise atas nama Anda. Jika sebelumnya Anda telah membuat peran layanan, AWS IoT SiteWise berikan daftar peran untuk dipilih. Jika tidak, Anda dapat membuat peran dengan izin yang diperlukan saat membuat portal. Penting untuk memilih peran yang memungkinkan akses ke aset dan data aset Anda. Untuk informasi selengkapnya, lihat [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#).

Otorisasi berdasarkan tanda AWS IoT SiteWise

Anda dapat melampirkan tag ke AWS IoT SiteWise sumber daya atau meneruskan tag dalam permintaan AWS IoT SiteWise. Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi

`aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`. Untuk informasi selengkapnya tentang penandaan sumber daya AWS IoT SiteWise, lihat [Tandai AWS IoT SiteWise sumber daya Anda](#).

Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke sumber daya berdasarkan tanda pada sumber daya tersebut, lihat [Lihat AWS IoT SiteWise aset berdasarkan tag](#).

AWS IoT SiteWise kebijakan berbasis identitas

Kebijakan IAM memungkinkan Anda mengontrol siapa yang dapat melakukan apa. AWS IoT SiteWise Anda dapat memutuskan tindakan apa yang diizinkan atau tidak dan menetapkan kondisi khusus untuk tindakan ini. Misalnya, Anda dapat membuat aturan tentang siapa yang dapat melihat atau mengubah informasi AWS IoT SiteWise. AWS IoT SiteWise mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan IAM JSON](#) dalam Panduan Pengguna IAM.

Tindakan kebijakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan AWS IoT SiteWise menggunakan awalan berikut sebelum tindakan:`iotsitewise:`. Misalnya, untuk memberikan izin kepada seseorang untuk mengunggah data properti aset AWS IoT SiteWise dengan operasi `BatchPutAssetPropertyValue` API, Anda menyertakan `iotsitewise:BatchPutAssetPropertyValue` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus mencakup salah satu `Action` atau `NotAction` elemen. AWS IoT SiteWise mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan tindakan dengan koma seperti berikut:

```
"Action": [  
  "iotsitewise:action1",  
  "iotsitewise:action2"  
]
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut.

```
"Action": "iotsitewise:Describe*"
```

Untuk melihat daftar AWS IoT SiteWise tindakan, lihat [Tindakan yang ditentukan oleh AWS IoT SiteWise](#) dalam Panduan Pengguna IAM.

BatchPutAssetPropertyValue otorisasi

AWS IoT SiteWise mengotorisasi akses ke [BatchPutAssetPropertyValuetindakan](#) dengan cara yang tidak biasa. Untuk sebagian besar tindakan, saat Anda mengizinkan atau menolak akses, tindakan tersebut mengembalikan kesalahan jika izin tidak diberikan.

Dengan `BatchPutAssetPropertyValue`, Anda dapat mengirim beberapa entri data ke aset dan properti aset yang berbeda dalam satu permintaan API. AWS IoT SiteWise mengotorisasi setiap entri data secara independen. Untuk setiap entri individu yang gagal otorisasi dalam permintaan, AWS IoT SiteWise termasuk `AccessDeniedException` dalam daftar kesalahan yang dikembalikan. AWS IoT SiteWise menerima data untuk setiap entri yang mengotorisasi dan berhasil, bahkan jika entri lain dalam permintaan yang sama gagal.

Important

Sebelum Anda menelan data ke aliran data, lakukan hal berikut:

- Otorisasi `time-series` sumber daya jika Anda menggunakan alias properti untuk mengidentifikasi aliran data.
- Otorisasi `asset` sumber daya jika Anda menggunakan ID aset untuk mengidentifikasi aset yang berisi properti aset terkait.

Sumber daya kebijakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*" 
```

Setiap pernyataan kebijakan IAM berlaku untuk sumber daya yang Anda tentukan dengan menggunakan ARNs. ARN memiliki sintaks umum berikut.

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

Untuk informasi selengkapnya tentang format ARNs, lihat [Mengidentifikasi AWS sumber daya dengan Nama Sumber Daya Amazon \(ARNs\)](#).

Misalnya, untuk menentukan aset dengan ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` dalam pernyataan Anda, gunakan ARN berikut. ;

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE" 
```

Untuk menentukan semua aliran data milik akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*" 
```

Untuk menentukan semua aset milik akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*" 
```

Beberapa AWS IoT SiteWise tindakan, seperti untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

```
"Resource": "*" 
```

Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARNs dengan koma.

```
"Resource": [
  "resource1",
  "resource2"
]
```

Untuk melihat daftar jenis AWS IoT SiteWise sumber daya dan jenis sumber daya ARNs, lihat [Jenis sumber daya yang ditentukan oleh AWS IoT SiteWise](#) dalam Panduan Pengguna IAM. Untuk mempelajari tindakan yang dapat menentukan ARN setiap sumber daya, lihat [Tindakan yang ditentukan oleh AWS IoT SiteWise](#).

Kunci kondisi kebijakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen Condition (atau blok Condition) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen Condition bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen Condition dalam sebuah pernyataan, atau beberapa kunci dalam elemen Condition tunggal, maka AWS akan mengevaluasinya menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM: variabel dan tanda](#) dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

⚠ Important

Beberapa kunci syarat bersifat khusus untuk sumber daya, dan beberapa tindakan API menggunakan beberapa sumber daya. Jika Anda menulis pernyataan kebijakan dengan kunci kondisi, gunakan `Resource` elemen pernyataan untuk menentukan sumber daya yang diterapkan kunci kondisi. Jika Anda tidak melakukannya, kebijakan tersebut dapat mencegah pengguna melakukan tindakan sama sekali, karena pemeriksaan kondisi gagal untuk sumber daya yang tidak diterapkan oleh kunci kondisi. Jika Anda tidak ingin menentukan sumber daya, atau jika Anda telah menulis `Action` elemen kebijakan untuk menyertakan beberapa tindakan API, Anda harus menggunakan tipe `...IfExists` kondisi untuk memastikan bahwa kunci kondisi diabaikan untuk sumber daya yang tidak menggunakannya. Untuk informasi lebih lanjut, lihat [...IfExists](#) kondisi dalam Panduan Pengguna IAM.

AWS IoT SiteWise mendefinisikan kumpulan kunci kondisinya sendiri dan juga mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

AWS IoT SiteWise kunci kondisi

Kunci syarat	Deskripsi	Tipe
<code>iotsitewise:isAssociatedWithAssetProperty</code>	Apakah aliran data dikaitkan dengan properti aset. Gunakan kunci kondisi ini untuk menentukan izin berdasarkan keberadaan properti aset terkait untuk aliran data. Nilai contoh: <code>true</code>	String
<code>iotsitewise:assetHierarchyPath</code>	Jalur hierarki aset, yang merupakan string aset yang IDs masing-masing dipisahkan oleh garis miring ke depan. Gunakan kunci kondisi ini untuk menentukan izin	String

Kunci syarat	Deskripsi	Tipe
	berdasarkan subset hierarki semua aset di akun Anda. Nilai contoh: /a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/a1b2c3d4-5678-90ab-cdef-6666EXAMPLE	
<code>iotsitewise:propertyId</code>	ID properti aset. Gunakan kunci kondisi ini untuk menentukan izin berdasarkan properti tertentu dari model aset. Kunci kondisi ini berlaku untuk semua aset model itu. Nilai contoh: a1b2c3d4-5678-90ab-cdef-3333EXAMPLE	String
<code>iotsitewise:childAssetId</code>	ID aset yang dikaitkan sebagai anak ke aset lain. Gunakan kunci kondisi ini untuk menentukan izin berdasarkan aset anak. Untuk menentukan izin berdasarkan aset induk, gunakan bagian sumber daya dari pernyataan kebijakan. Nilai contoh: a1b2c3d4-5678-90ab-cdef-6666EXAMPLE	String

Kunci syarat	Deskripsi	Tipe
<code>iotsitewise:iam</code>	ARN identitas IAM saat mencantumkan kebijakan akses. Gunakan kunci kondisi ini untuk menentukan izin kebijakan akses untuk identitas IAM. Nilai contoh: <code>arn:aws:iam::123456789012:user/JohnDoe</code>	Tali, Null
<code>iotsitewise:propertyAlias</code>	Alias yang mengidentifikasi properti aset atau aliran data. Gunakan tombol kondisi ini untuk menentukan izin berdasarkan alias.	String
<code>iotsitewise:user</code>	ID pengguna Pusat Identitas IAM saat mencantumkan kebijakan akses. Gunakan kunci kondisi ini untuk menentukan izin kebijakan akses bagi pengguna Pusat Identitas IAM. Nilai contoh: <code>a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE</code>	Tali, Null

Kunci syarat	Deskripsi	Tipe
<code>iotsitewise:group</code>	ID grup Pusat Identitas IAM saat mencantumkan kebijakan akses. Gunakan kunci kondisi ini untuk menentukan izin kebijakan akses untuk grup Pusat Identitas IAM. Nilai contoh: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE	Tali, Null
<code>iotsitewise:portal</code>	ID portal dalam kebijakan akses. Gunakan kunci kondisi ini untuk menentukan izin kebijakan akses berdasarkan portal. Nilai contoh: a1b2c3d4-5678-90ab-cdef-7777EXAMPLE	Tali, Null
<code>iotsitewise:project</code>	ID proyek dalam kebijakan akses, atau ID proyek untuk dasbor. Gunakan tombol kondisi ini untuk menentukan dasbor atau akses izin kebijakan berdasarkan proyek. Nilai contoh: a1b2c3d4-5678-90ab-cdef-8888EXAMPLE	Tali, Null

Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh AWS IoT SiteWise](#).

Contoh

Untuk melihat contoh kebijakan AWS IoT SiteWise berbasis identitas, lihat. [AWS IoT SiteWise contoh kebijakan berbasis identitas](#)

AWS IoT SiteWise contoh kebijakan berbasis identitas

Secara default, entitas (pengguna dan peran) tidak memiliki izin untuk membuat atau memodifikasi AWS IoT SiteWise sumber daya. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk menyesuaikan izin, administrator AWS Identity and Access Management (IAM) harus melakukan hal berikut:

1. Buat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang mereka butuhkan.
2. Lampirkan kebijakan tersebut ke pengguna atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan pada tab JSON](#) dalam Panduan Pengguna IAM.

Topik

- [Praktik terbaik kebijakan](#)
- [Gunakan AWS IoT SiteWise konsol](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)
- [Izinkan pengguna untuk menelan data ke aset dalam satu hierarki](#)
- [Lihat AWS IoT SiteWise aset berdasarkan tag](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS IoT SiteWise sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola

yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [Kebijakan yang dikelola AWS](#) atau [Kebijakan yang dikelola AWS untuk fungsi tugas](#) dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [Validasi kebijakan dengan IAM Access Analyzer](#) dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Amankan akses API dengan MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

Gunakan AWS IoT SiteWise konsol

Untuk mengakses AWS IoT SiteWise konsol, Anda memerlukan satu set izin dasar. Izin ini memungkinkan Anda melihat dan mengelola detail tentang AWS IoT SiteWise sumber daya di AWS akun Anda.

Jika Anda membuat kebijakan yang terlalu ketat, konsol mungkin tidak berfungsi seperti yang diharapkan untuk pengguna atau peran (entitas) dengan kebijakan tersebut. Untuk memastikan bahwa entitas tersebut masih dapat menggunakan AWS IoT SiteWise konsol, lampirkan kebijakan [AWSIoTSiteWiseConsoleFullAccess](#) terkelola ke mereka atau tentukan izin yang setara untuk entitas tersebut. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) dalam Panduan Pengguna IAM.

Jika entitas hanya menggunakan AWS Command Line Interface (CLI) atau AWS IoT SiteWise API, dan bukan konsol, mereka tidak memerlukan izin minimum ini. Dalam hal ini, cukup beri mereka akses ke tindakan spesifik yang mereka butuhkan untuk tugas API mereka.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "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": "*"
}
]
}

```

Izinkan pengguna untuk menelan data ke aset dalam satu hierarki

Dalam contoh ini, Anda ingin memberi pengguna di AWS akun Anda akses untuk menulis data ke semua properti aset dalam hierarki aset tertentu, mulai dari aset `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` root. Kebijakan memberikan `iotsitewise:BatchPutAssetPropertyValue` izin kepada pengguna. Kebijakan ini menggunakan kunci `iotsitewise:assetHierarchyPath` kondisi untuk membatasi akses ke aset yang jalur hierarkinya cocok dengan aset atau turunannya.

```

{
  "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/*"
          ]
        }
      }
    }
  ]
}

```



```
}
```

Lihat AWS IoT SiteWise aset berdasarkan tag

Gunakan kondisi dalam kebijakan berbasis identitas Anda untuk mengontrol akses ke AWS IoT SiteWise sumber daya berdasarkan tag. Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan penayangan aset. Namun, izin diberikan hanya jika tag aset `Owner` memiliki nilai nama pengguna pengguna tersebut. Kebijakan ini juga memberikan izin untuk menyelesaikan tindakan ini di konsol.

```
{
  "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}"
        }
      }
    }
  ]
}
```

Lampirkan kebijakan ini ke pengguna di akun Anda. Jika pengguna bernama `richard-roe` mencoba untuk melihat AWS IoT SiteWise aset, aset harus ditandai `Owner=richard-roe` atau `owner=richard-roe`. Jika tidak, Richard ditolak aksesnya. Nama kunci tag kondisi tidak peka huruf besar/kecil. Jadi, `Owner` cocok dengan keduanya `Owner` dan `owner`. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM JSON: Syarat](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan di AWS IoT SiteWise

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat [Pilih antara kebijakan yang dikelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACLs. Untuk mempelajari selengkapnya ACLs, lihat [Ringkasan daftar kontrol akses \(ACL\)](#) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations

adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat [Kebijakan kontrol layanan](#) di Panduan AWS Organizations Pengguna.

- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat [Kebijakan kontrol sumber daya \(RCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

AWS kebijakan terkelola untuk AWS IoT SiteWise

Sederhanakan menambahkan izin ke pengguna, grup, dan peran menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk [membuat kebijakan terkelola pelanggan IAM](#) yang memberikan izin yang tepat kepada tim Anda. Untuk penyiapan yang lebih cepat, pertimbangkan untuk menggunakan kebijakan AWS terkelola kami untuk kasus penggunaan umum. Temukan kebijakan AWS terkelola di AWS akun Anda. Untuk informasi lebih lanjut tentang kebijakan yang dikelola AWS, lihat [kebijakan yang dikelola AWS](#) di Panduan Pengguna IAM.

AWS Layanan mengurus pembaruan dan pemeliharaan kebijakan AWS terkelola, artinya Anda tidak dapat mengubah izin kebijakan ini. Terkadang, AWS IoT SiteWise dapat menambahkan izin untuk mengakomodasi fitur baru, yang memengaruhi semua identitas dengan kebijakan yang dilampirkan. Pembaruan semacam itu biasa terjadi pada pengenalan layanan atau fitur baru. Namun, izin tidak pernah dihapus, memastikan pengaturan Anda tetap utuh.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS menambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk daftar dengan deskripsi kebijakan fungsi pekerjaan, lihat [kebijakan AWS terkelola untuk fungsi pekerjaan](#) di Panduan Pengguna IAM.

AWS kebijakan terkelola: `AWSIoTSiteWiseReadOnlyAccess`

Gunakan kebijakan `AWSIoTSiteWiseReadOnlyAccess` AWS terkelola untuk mengizinkan akses hanya-baca. AWS IoT SiteWise

Anda dapat melampirkan kebijakan `AWSIoTSiteWiseReadOnlyAccess` ke identitas IAM Anda.

Izin tingkat layanan

Kebijakan ini menyediakan akses hanya-baca ke. AWS IoT SiteWise Tidak ada izin layanan lain yang disertakan dalam kebijakan ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:BatchGet*",
        "iotsitewise:Get*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS kebijakan terkelola: AWSService RoleForIo TSite Bijaksana

AWSServiceRoleForIoTSiteWisePeran menggunakan AWSServiceRoleForIoTSiteWise kebijakan dengan izin berikut. Kebijakan ini:

- Memungkinkan AWS IoT SiteWise untuk menyebarkan gateway SiteWise Edge (yang berjalan).
AWS IoT Greengrass
- Memungkinkan AWS IoT SiteWise untuk melakukan logging.
- Memungkinkan AWS IoT SiteWise untuk menjalankan kueri pencarian metadata, terhadap database. AWS IoT TwinMaker

Jika Anda menggunakan AWS IoT SiteWise akun pengguna singe, AWSServiceRoleForIoTSiteWise peran akan membuat AWSServiceRoleForIoTSiteWise kebijakan di akun IAM Anda, dan melampirkannya ke peran terkait AWSServiceRoleForIoTSiteWise [Layanan](#) untuk. 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"
      ]
    }
  }
}
]
}

```

AWS IoT SiteWise pembaruan kebijakan AWS terkelola

Anda dapat melihat detail tentang pembaruan kebijakan AWS terkelola untuk AWS IoT SiteWise, mulai dari saat layanan ini mulai melacak perubahan. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat AWS IoT SiteWise dokumen.

Perubahan	Deskripsi	Tanggal
AWSServiceRoleForIoTSiteBijaksana - Perbarui ke kebijakan yang ada	AWS IoT SiteWise sekarang dapat menjalankan kueri pencarian metadata,	6 November 2023

Perubahan	Deskripsi	Tanggal
	terhadap database. AWS IoT TwinMaker	
AWSIoTSiteWiseReadOnlyAccess – Pembaruan ke kebijakan yang ada	AWS IoT SiteWise menambahkan awalan kebijakan baru, BatchGet*, yang memungkinkan Anda melakukan operasi pembacaan batch.	September 16, 2022
AWSIoTSiteWiseReadOnlyAccess – Kebijakan baru	AWS IoT SiteWise menambahkan kebijakan baru untuk memberikan akses hanya-baca ke. AWS IoT SiteWise	24 November 2021
AWS IoT SiteWise mulai melacak perubahan	AWS IoT SiteWise mulai melacak perubahan untuk kebijakan AWS terkelolanya.	24 November 2021

Gunakan peran terkait layanan untuk AWS IoT SiteWise

AWS IoT SiteWise menggunakan AWS Identity and Access Management peran [terkait layanan](#) (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke AWS IoT SiteWise. Peran terkait layanan telah ditentukan sebelumnya oleh AWS IoT SiteWise dan menyertakan semua izin yang diperlukan layanan untuk memanggil layanan lain atas nama Anda. AWS

Peran terkait layanan menyederhanakan konfigurasi AWS IoT SiteWise dengan secara otomatis menyertakan semua izin yang diperlukan. AWS IoT SiteWise mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya AWS IoT SiteWise dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin. Dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi AWS IoT SiteWise sumber daya Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat [AWS layanan yang bekerja dengan IAM](#) dan cari layanan yang memiliki Ya di kolom Peran Tertaut Layanan. Pilih Ya dengan tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Topik

- [Izin peran terkait layanan untuk AWS IoT SiteWise](#)
- [Membuat peran terkait layanan untuk AWS IoT SiteWise](#)
- [Memperbarui peran terkait layanan untuk AWS IoT SiteWise](#)
- [Menghapus peran terkait layanan untuk AWS IoT SiteWise](#)
- [Wilayah yang Didukung untuk AWS IoT SiteWise peran terkait layanan](#)
- [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#)

Izin peran terkait layanan untuk AWS IoT SiteWise

AWS IoT SiteWise menggunakan peran terkait layanan bernama `AWSServiceRoleForIoTSiteWise`. AWS IoT SiteWise menggunakan peran terkait layanan ini untuk menerapkan gateway SiteWise Edge (yang berjalan) dan melakukan logging. AWS IoT Greengrass

Peran `AWSServiceRoleForIoTSiteWise` terkait layanan menggunakan `AWSServiceRoleForIoTSiteWise` kebijakan dengan izin berikut. Kebijakan ini:

- Memungkinkan AWS IoT SiteWise untuk menyebarkan gateway SiteWise Edge (yang berjalan).
AWS IoT Greengrass
- Memungkinkan AWS IoT SiteWise untuk melakukan logging.
- Memungkinkan AWS IoT SiteWise untuk menjalankan kueri pencarian metadata, terhadap database. AWS IoT TwinMaker

Untuk informasi selengkapnya tentang tindakan yang diizinkan `AWSServiceRoleForIoTSiteWise`, lihat [kebijakan AWS terkelola untuk 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"
      ]
    }
  }
}

```

```
}  
}  
}  
]  
}
```

Anda dapat menggunakan log untuk memantau dan memecahkan masalah gateway SiteWise Edge Anda. Untuk informasi selengkapnya, lihat [Pantau log gateway SiteWise Edge](#).

Untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) membuat, mengedit, atau menghapus peran terkait layanan, pertama-tama konfigurasi izin. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk AWS IoT SiteWise

AWS IoT SiteWise memerlukan peran terkait layanan untuk melakukan tindakan tertentu dan mengakses sumber daya atas nama Anda. Peran terkait layanan adalah jenis unik dari peran AWS Identity and Access Management (IAM) dan Access Management (IAM) yang ditautkan langsung ke AWS IoT SiteWise. Dengan membuat peran ini, Anda memberikan izin AWS IoT SiteWise yang diperlukan untuk mengakses AWS layanan dan sumber daya lain yang diperlukan untuk operasinya, seperti Amazon S3 untuk penyimpanan data AWS IoT atau untuk komunikasi perangkat.

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda melakukan operasi berikut di AWS IoT SiteWise konsol, AWS IoT SiteWise buat peran terkait layanan untuk Anda.

- Buat gateway Greengrass V1.
- Konfigurasi opsi logging.
- Memilih tombol keikutsertaan di spanduk eksekusi kueri.

Jika Anda menghapus peran terkait layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda melakukan operasi apa pun di AWS IoT SiteWise konsol, AWS IoT SiteWise buat peran terkait layanan untuk Anda lagi.

Anda juga dapat menggunakan konsol IAM atau API untuk membuat peran terkait layanan. AWS IoT SiteWise

- Untuk melakukannya di konsol IAM, buat peran dengan kebijakan `AWSServiceRoleForIoTSiteWise` dan hubungan kepercayaan dengan `aniotsitewise.amazonaws.com`.

- Untuk melakukannya menggunakan API AWS CLI atau IAM, buat peran dengan `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` kebijakan dan hubungan kepercayaan dengan `aniotsitewise.amazonaws.com`.

Untuk informasi selengkapnya, lihat [Membuat peran terkait layanan di Panduan Pengguna IAM](#).

Jika Anda menghapus peran terkait layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Memperbarui peran terkait layanan untuk AWS IoT SiteWise

AWS IoT SiteWise tidak memungkinkan Anda untuk mengedit peran terkait layanan `AWSServiceRoleForIoTSiteWise`. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Memperbarui peran terkait layanan](#) di Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk AWS IoT SiteWise

Jika fitur atau layanan yang memerlukan peran terkait layanan tidak lagi digunakan, disarankan untuk menghapus peran terkait. Ini untuk menghindari entitas tidak aktif yang tidak dipantau atau dipelihara. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

Note

Jika AWS IoT SiteWise layanan menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal tersebut terjadi, tunggu beberapa menit dan coba lagi.

Untuk menghapus AWS IoT SiteWise sumber daya yang digunakan oleh `AWSServiceRoleForIoTSiteWise`

1. Nonaktifkan logging untuk AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Ubah level logging Anda](#)
2. Hapus gateway SiteWise Edge yang aktif.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran terkait layanan AWSService RoleForIoT Site Wise. Untuk informasi selengkapnya, lihat [Menghapus peran atau profil instans](#) di Panduan Pengguna IAM.

Wilayah yang Didukung untuk AWS IoT SiteWise peran terkait layanan

AWS IoT SiteWise mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi lebih lanjut, lihat [Kuota dan Titik Akhir AWS IoT SiteWise](#).

Gunakan peran layanan untuk AWS IoT SiteWise Monitor

Peran layanan adalah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Untuk memungkinkan pengguna portal SiteWise Monitor federasi mengakses sumber daya Anda AWS IoT SiteWise dan AWS IAM Identity Center sumber daya, Anda harus melampirkan peran layanan ke setiap portal yang Anda buat. Peran layanan harus menentukan SiteWise Monitor sebagai entitas tepercaya dan menyertakan kebijakan [AWSIoTSiteWiseMonitorPortalAccess](#) terkelola atau menentukan [izin yang setara](#). Kebijakan ini dikelola oleh AWS dan menentukan kumpulan izin yang digunakan SiteWise Monitor untuk mengakses sumber daya Pusat Identitas IAM Anda AWS IoT SiteWise dan IAM.

Saat membuat portal SiteWise Monitor, Anda harus memilih peran yang memungkinkan pengguna portal tersebut mengakses sumber daya Pusat Identitas IAM AWS IoT SiteWise dan IAM Anda. AWS IoT SiteWise Konsol dapat membuat dan mengonfigurasi peran untuk Anda. Anda dapat mengedit peran di IAM nanti. Pengguna portal Anda akan mengalami masalah dalam menggunakan portal SiteWise Monitor mereka jika Anda menghapus izin yang diperlukan dari peran atau menghapus peran tersebut.

Note

Portal yang dibuat sebelum 29 April 2020 tidak memerlukan peran layanan. Jika Anda membuat portal sebelum tanggal ini, Anda harus melampirkan peran layanan untuk terus menggunakannya. Untuk melakukannya, navigasikan ke halaman Portal di [AWS IoT SiteWise konsol](#), lalu pilih Migrasi semua portal untuk menggunakan peran IAM.

Bagian berikut menjelaskan cara membuat dan mengelola peran layanan SiteWise Monitor di AWS Management Console atau AWS Command Line Interface.

Daftar Isi

- [Izin peran layanan untuk SiteWise Monitor \(Klasik\)](#)
- [Izin peran layanan untuk SiteWise Monitor \(AI-aware\)](#)
- [Mengelola peran layanan SiteWise Monitor \(konsol\)](#)
 - [Temukan peran layanan portal \(konsol\)](#)
 - [Membuat peran layanan SiteWise Monitor \(AWS IoT SiteWise konsol\)](#)
 - [Membuat peran layanan SiteWise Monitor \(konsol IAM\)](#)
 - [Mengubah peran layanan portal \(konsol\)](#)
- [Mengelola peran layanan SiteWise Monitor \(CLI\)](#)
 - [Temukan peran layanan portal \(CLI\)](#)
 - [Buat peran layanan SiteWise Monitor \(CLI\)](#)
- [SiteWise Pantau pembaruan ke AWSIoTSiteWiseMonitorServiceRole](#)

Izin peran layanan untuk SiteWise Monitor (Klasik)

Saat membuat portal, AWS IoT SiteWise Anda dapat membuat peran yang namanya dimulai AWSIoTSiteWiseMonitorServiceRole. Peran ini memungkinkan pengguna SiteWise Monitor federasi untuk mengakses konfigurasi portal, aset, data aset, dan konfigurasi Pusat Identitas IAM Anda.

Peran tersebut mempercayai layanan berikut untuk mengambil peran:

- `monitor.iotsitewise.amazonaws.com`

Peran tersebut menggunakan kebijakan izin berikut, yang dimulai dengan AWSIoTSiteWiseMonitorServicePortalPolicy, untuk memungkinkan pengguna SiteWise Monitor menyelesaikan tindakan pada sumber daya di akun Anda. Kebijakan [AWSIoTSiteWiseMonitorPortalAccess](#) terkelola mendefinisikan izin yang setara.

```
{
  "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"
            ]
        }
    }
}
]
}

```

Untuk informasi selengkapnya tentang izin yang diperlukan untuk alarm, lihat. [Siapkan izin untuk alarm acara di AWS IoT SiteWise](#)

Saat pengguna portal masuk, SiteWise Monitor akan membuat [kebijakan sesi](#) berdasarkan persimpangan peran layanan dan kebijakan akses pengguna tersebut. Kebijakan akses menentukan tingkat akses identitas ke portal dan proyek Anda. Untuk informasi selengkapnya tentang izin portal dan kebijakan akses, lihat [Mengelola portal SiteWise Monitor Anda](#) dan [CreateAccessPolicy](#).

Izin peran layanan untuk SiteWise Monitor (AI-aware)

Saat Anda membuat portal, AWS IoT SiteWise memungkinkan Anda membuat peran yang namanya dimulai dengan `IoT Site WisePortalRole`. Peran ini memungkinkan pengguna SiteWise Monitor federasi untuk mengakses konfigurasi portal, aset, data aset, dan konfigurasi Pusat Identitas IAM Anda.

Warning

Pemilik proyek dan peran penampil Proyek tidak didukung untuk SiteWise Monitor (sadar AI).

Peran tersebut mempercayai layanan berikut untuk mengambil peran:

- `monitor.iotsitewise.amazonaws.com`

Peran tersebut menggunakan kebijakan izin berikut, yang dimulai dengan `IoT Site Wise AIPortal AccessPolicy`, untuk memungkinkan pengguna SiteWise Monitor menyelesaikan tindakan pada sumber daya di akun Anda.

```

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

```

```
{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:CreateProject",
    "iotsitewise:DescribePortal",
    "iotsitewise:ListProjects",
    "iotsitewise:DescribeProject",
    "iotsitewise:UpdateProject",
    "iotsitewise>DeleteProject",
    "iotsitewise:CreateDashboard",
    "iotsitewise:DescribeDashboard",
    "iotsitewise:UpdateDashboard",
    "iotsitewise>DeleteDashboard",
    "iotsitewise:ListDashboards",
    "iotsitewise:ListAssets",
    "iotsitewise:DescribeAsset",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:ListAssetProperties",
    "iotsitewise:DescribeAssetProperty",
    "iotsitewise:GetAssetPropertyValue",
    "iotsitewise:GetAssetPropertyValueHistory",
    "iotsitewise:GetAssetPropertyAggregates",
    "iotsitewise:GetInterpolatedAssetPropertyValues",
    "iotsitewise:BatchGetAssetPropertyAggregates",
    "iotsitewise:BatchGetAssetPropertyValue",
    "iotsitewise:BatchGetAssetPropertyValueHistory",
    "iotsitewise:ListAssetRelationships",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:ListAssetModels",
    "iotsitewise:DescribeAssetCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
    "iotsitewise:ListAssetModelProperties",
    "iotsitewise:ExecuteQuery",
    "iotsitewise:ListTimeSeries",
    "iotsitewise:DescribeTimeSeries",
    "iotsitewise:InvokeAssistant",
    "iotsitewise:DescribeDataset",
    "iotsitewise:ListDatasets",
    "iotevents:DescribeAlarmModel",
    "iotevents:ListTagsForResource",
    "iottwinmaker:ListWorkspaces",
    "iottwinmaker:ExecuteQuery",
    "iottwinmaker:GetWorkspace",
    "identitystore:DescribeUser"
```

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

Saat pengguna portal masuk, SiteWise Monitor akan membuat [kebijakan sesi](#) berdasarkan persimpangan peran layanan dan kebijakan akses pengguna tersebut.

Mengelola peran layanan SiteWise Monitor (konsol)

Konsol AWS IoT SiteWise Memfasilitasi pengelolaan peran layanan SiteWise Monitor untuk portal. Setelah membuat portal, konsol memeriksa peran yang ada yang cocok untuk lampiran. Jika tidak ada yang tersedia, konsol dapat membuat dan mengonfigurasi peran layanan untuk Anda. Untuk informasi selengkapnya, lihat [Buat portal di SiteWise Monitor](#).

Topik

- [Temukan peran layanan portal \(konsol\)](#)
- [Membuat peran layanan SiteWise Monitor \(AWS IoT SiteWise konsol\)](#)
- [Membuat peran layanan SiteWise Monitor \(konsol IAM\)](#)
- [Mengubah peran layanan portal \(konsol\)](#)

Temukan peran layanan portal (konsol)

Gunakan langkah-langkah berikut untuk menemukan peran layanan yang dilampirkan ke portal SiteWise Monitor.

Untuk menemukan peran layanan portal

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Portal.
3. Pilih portal yang ingin Anda temukan peran layanannya.

Peran yang dilampirkan ke portal muncul di bawah Izin, peran Layanan.

Membuat peran layanan SiteWise Monitor (AWS IoT SiteWise konsol)

Saat membuat portal SiteWise Monitor, Anda dapat membuat peran layanan untuk portal Anda. Untuk informasi selengkapnya, lihat [Buat portal di SiteWise Monitor](#).

Anda juga dapat membuat peran layanan untuk portal yang ada di AWS IoT SiteWise konsol. Ini menggantikan peran layanan portal yang ada.

Untuk membuat peran layanan untuk portal yang ada

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Portal.
3. Pilih portal yang ingin Anda buat peran layanan baru.
4. Di bawah Detail portal, pilih Edit.
5. Di bawah Izin, pilih Buat dan gunakan peran layanan baru dari daftar.
6. Masukkan nama untuk peran baru Anda.
7. Pilih Simpan.

Membuat peran layanan SiteWise Monitor (konsol IAM)

Anda dapat membuat peran layanan dari templat peran layanan di konsol IAM. Templat peran ini mencakup kebijakan [AWSIoTSiteWiseMonitorPortalAccess](#)sterkelola dan menetapkan SiteWise Monitor sebagai entitas tepercaya.

Untuk membuat peran layanan dari template peran layanan portal

1. Arahkan ke [konsol IAM](#).
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Dalam Pilih kasus penggunaan, pilih IoT SiteWise.
5. Di Pilih kasus penggunaan Anda, pilih IoT SiteWise Monitor - Portal.
6. Pilih Berikutnya: Izin.
7. Pilih Berikutnya: Tanda.
8. Pilih Berikutnya: Tinjau.
9. Masukkan nama Peran untuk peran layanan baru.
10. Pilih Buat peran.

Mengubah peran layanan portal (konsol)

Gunakan prosedur berikut untuk memilih peran layanan SiteWise Monitor yang berbeda untuk portal.

Untuk mengubah peran layanan portal

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih Portal.
3. Pilih portal yang ingin Anda ubah peran layanannya.
4. Di bawah Detail portal, pilih Edit.
5. Di bawah Izin, pilih Gunakan peran yang ada.
6. Pilih peran yang ada untuk dilampirkan ke portal ini.
7. Pilih Simpan.

Mengelola peran layanan SiteWise Monitor (CLI)

Anda dapat menggunakan AWS CLI untuk tugas manajemen peran layanan portal berikut:

Topik

- [Temukan peran layanan portal \(CLI\)](#)
- [Buat peran layanan SiteWise Monitor \(CLI\)](#)

Temukan peran layanan portal (CLI)

Untuk menemukan peran layanan yang dilampirkan ke portal SiteWise Monitor, jalankan perintah berikut untuk mencantumkan semua portal Anda di Wilayah saat ini.

```
aws iotsitewise list-portals
```

Operasi mengembalikan respons yang berisi ringkasan portal Anda dalam format berikut.

```
{
  "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"
    }
  ]
}
```

```
]
}
```

Anda juga dapat menggunakan [DescribePortal](#) operasi untuk menemukan peran portal Anda jika Anda mengetahui ID portal Anda.

Buat peran layanan SiteWise Monitor (CLI)

Gunakan langkah-langkah berikut untuk membuat peran layanan SiteWise Monitor baru.

Untuk membuat peran layanan SiteWise Monitor

1. Buat peran dengan kebijakan kepercayaan yang memungkinkan SiteWise Monitor untuk mengambil peran. Contoh ini membuat peran bernama **MySiteWiseMonitorPortalRole** dari kebijakan kepercayaan yang disimpan dalam string JSON.

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. Salin peran ARN dari metadata peran dalam output. Saat Anda membuat portal, Anda menggunakan ARN ini untuk mengaitkan peran dengan portal Anda. Untuk informasi selengkapnya tentang membuat portal, lihat [CreatePortal](#) di Referensi AWS IoT SiteWise API.

3. a. Untuk SiteWise Monitor (Klasik) — Lampirkan `AWSIoTSiteWiseMonitorPortalAccess` kebijakan ke peran, atau lampirkan kebijakan yang menentukan izin yang setara.

```
aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess
```

- b. Untuk SiteWise Monitor (AI-aware) — Lampirkan `IoTSiteWiseAIPortalAccessPolicy` kebijakan ke peran, atau lampirkan kebijakan yang menentukan izin yang setara. Misalnya, buat kebijakan dengan izin akses portal. Contoh berikut membuat kebijakan bernama `MySiteWiseMonitorPortalAccess`.

```
aws iam create-policy \
  --policy-name MySiteWiseMonitorPortalAccess \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iotsitewise:CreateProject",
          "iotsitewise:DescribePortal",
          "iotsitewise:ListProjects",
          "iotsitewise:DescribeProject",
          "iotsitewise:UpdateProject",
          "iotsitewise>DeleteProject",
          "iotsitewise:CreateDashboard",
          "iotsitewise:DescribeDashboard",
          "iotsitewise:UpdateDashboard",
          "iotsitewise>DeleteDashboard",
          "iotsitewise:ListDashboards",
          "iotsitewise:ListAssets",
          "iotsitewise:DescribeAsset",
          "iotsitewise:ListAssociatedAssets",
          "iotsitewise:ListAssetProperties",
          "iotsitewise:DescribeAssetProperty",
          "iotsitewise:GetAssetPropertyValue",
          "iotsitewise:GetAssetPropertyValueHistory",
          "iotsitewise:GetAssetPropertyAggregates",
          "iotsitewise:GetInterpolatedAssetPropertyValues",
          "iotsitewise:BatchGetAssetPropertyAggregates",
          "iotsitewise:BatchGetAssetPropertyValue",
```

```

        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:DescribeAssetCompositeModel",
        "iotsitewise:DescribeAssetModelCompositeModel",
        "iotsitewise:ListAssetModelProperties",
        "iotsitewise:ExecuteQuery",
        "iotsitewise:ListTimeSeries",
        "iotsitewise:DescribeTimeSeries",
        "iotsitewise:InvokeAssistant",
        "iotsitewise:DescribeDataset",
        "iotsitewise:ListDatasets",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource",
        "iottwinmaker:ListWorkspaces",
        "iottwinmaker:ExecuteQuery",
        "iottwinmaker:GetWorkspace",
        "identitystore:DescribeUser"
    ],
    "Resource": "*"
}
]
}'

```

Untuk melampirkan peran layanan ke portal yang ada

1. Untuk mengambil detail portal yang ada, jalankan perintah berikut. Ganti *portal-id* dengan ID portal.

```
aws iotsitewise describe-portal --portal-id portal-id
```

Operasi mengembalikan respons yang berisi rincian portal dalam format berikut.

```

{
  "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. Untuk melampirkan peran layanan ke portal, jalankan perintah berikut. Ganti *role-arn* dengan ARN peran layanan, dan ganti parameter yang tersisa dengan nilai portal yang ada.

```

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 Pantau pembaruan ke AWS IoT SiteWise MonitorServiceRole

Anda dapat melihat detail tentang pembaruan AWS IoT SiteWise MonitorServiceRole untuk SiteWise Monitor, mulai dari saat layanan ini mulai melacak perubahan. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat AWS IoT SiteWise dokumen.

Perubahan	Deskripsi	Tanggal
AWS IoT SiteWise Monitor Portal Access — Kebijakan yang diperbarui	AWS IoT SiteWise memperbarui kebijakan AWS IoT SiteWise Monitor Portal Access terkelola untuk fitur alarm.	27 Mei 2021

Perubahan	Deskripsi	Tanggal
AWS IoT SiteWise mulai melacak perubahan	AWS IoT SiteWise mulai melacak perubahan untuk peran layanannya.	15 Desember 2020

Siapkan izin untuk alarm acara di AWS IoT SiteWise

Bila Anda menggunakan model AWS IoT Events alarm untuk memantau properti AWS IoT SiteWise aset, Anda harus memiliki izin IAM berikut:

- Peran AWS IoT Events layanan yang memungkinkan AWS IoT Events untuk mengirim data ke AWS IoT SiteWise. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses AWS IoT Events](#) di Panduan AWS IoT Events Pengembang.
- Anda harus memiliki izin AWS IoT SiteWise tindakan berikut:
`iotsitewise:DescribeAssetModel`
`daniotsitewise:UpdateAssetModelPropertyRouting`. Izin ini memungkinkan AWS IoT SiteWise untuk mengirim nilai properti aset ke model AWS IoT Events alarm.

Untuk informasi selengkapnya, lihat [Kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Izin tindakan yang diperlukan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa. Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan.

Sebelum Anda menentukan model AWS IoT Events alarm, Anda harus memberikan izin berikut yang memungkinkan AWS IoT SiteWise untuk mengirim nilai properti aset ke model alarm.

- `iotsitewise:DescribeAssetModel`, `iotsitewise:ListAssetModels` — Memungkinkan AWS IoT Events untuk memeriksa apakah properti aset ada.
- `iotsitewise:UpdateAssetModelPropertyRouting` — Memungkinkan AWS IoT SiteWise untuk secara otomatis membuat langganan yang memungkinkan AWS IoT SiteWise untuk mengirim data ke AWS IoT Events.

Untuk informasi selengkapnya tentang tindakan yang AWS IoT SiteWise didukung, lihat [Tindakan yang ditentukan oleh AWS IoT SiteWise](#) dalam Referensi Otorisasi Layanan.

Example Contoh kebijakan izin 1

Kebijakan berikut memungkinkan AWS IoT SiteWise untuk mengirim nilai properti aset ke model AWS IoT Events alarm apa pun.

```
{
  "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:ListAssetModels",
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

Example Contoh kebijakan izin 2

Kebijakan berikut memungkinkan AWS IoT SiteWise untuk mengirim nilai properti aset tertentu ke model AWS IoT Events alarm tertentu.

```
{
  "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:ListAssetModels"
    ],
    "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"
        }
    }
}
]
}

```

ListInputRoutings Izin (Opsional)

Saat Anda memperbarui atau menghapus model aset, AWS IoT SiteWise dapat memeriksa apakah model alarm di AWS IoT Events memantau properti aset yang terkait dengan model aset ini. Ini mencegah Anda menghapus properti aset yang saat ini digunakan AWS IoT Events alarm. Untuk mengaktifkan fitur ini AWS IoT SiteWise, Anda harus memiliki `iotevents:ListInputRoutings` izin. Izin ini memungkinkan AWS IoT SiteWise untuk melakukan panggilan ke operasi

[ListInputRoutings](#) API yang didukung oleh AWS IoT Events.

Note

Kami sangat menyarankan Anda menambahkan `ListInputRoutings` izin.

Example Contoh kebijakan izin

Kebijakan berikut memungkinkan Anda memperbarui dan menghapus model aset, dan menggunakan `ListInputRoutings` API di dalamnya 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/*"
    }
  ]
}
```

Izin yang diperlukan untuk Monitor SiteWise

Jika ingin menggunakan fitur alarm di portal SiteWise Monitor, Anda harus memperbarui [peran layanan SiteWise Monitor](#) dengan kebijakan berikut:

```
{
  "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"
        ]
      }
    }
  }
}

```

```
]
}
```

Cross-service membingungkan deputi pencegahan di AWS IoT SiteWise

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi [aws:SourceAccount](#) global [aws:SourceArn](#) dan dalam kebijakan sumber daya untuk membatasi izin yang AWS IoT SiteWise memberikan layanan lain ke sumber daya. Jika `aws:SourceArn` nilai tidak berisi ID akun, seperti bucket Amazon S3 Nama Sumber Daya Amazon (ARN), Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

- Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan.
- Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Nilai `aws:SourceArn` harus menjadi sumber daya AWS IoT SiteWise pelanggan yang terkait dengan `sts:AssumeRole` permintaan.

Cara paling efektif untuk melindungi dari masalah "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:service*:*:123456789012:*`.

Example — Pencegahan Deputi Bingung

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan AWS IoT SiteWise untuk mencegah masalah wakil yang membingungkan.

```
{
  "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"
      }
    }
  }
}
```

Memecahkan masalah AWS IoT SiteWise identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS IoT SiteWise dan AWS Identity and Access Management (IAM).

Topik

- [Saya tidak berwenang untuk melakukan tindakan di AWS IoT SiteWise](#)
- [Saya tidak berwenang untuk melakukan iam:PassRole](#)
- [Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS IoT SiteWise sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di AWS IoT SiteWise

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberikan nama pengguna dan kata sandi Anda.

Contoh kesalahan berikut terjadi ketika pengguna `mateojackson` IAM mencoba menggunakan konsol untuk melihat detail tentang aset tetapi tidak memiliki `iotsitewise:DescribeAsset` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotsitewise:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk memungkinkannya mengakses sumber daya aset dengan ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` menggunakan `iotsitewise:DescribeAsset` tindakan tersebut.

Saya tidak berwenang untuk melakukan **iam:PassRole**

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS IoT SiteWise.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di AWS IoT SiteWise. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS IoT SiteWise sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah AWS IoT SiteWise mendukung fitur-fitur ini, lihat [Bagaimana AWS IoT SiteWise bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM](#).

Validasi kepatuhan untuk AWS IoT SiteWise

AWS IoT SiteWise tidak dalam lingkup program AWS kepatuhan apa pun.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#). Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS IoT SiteWise ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan Panduan](#) Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- [Arsitektur untuk Whitepaper Keamanan dan Kepatuhan HIPAA — Whitepaper](#) ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi sumber daya dengan aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.
- [Sepuluh aturan emas keamanan untuk solusi IoT Industri](#) - Posting blog ini memperkenalkan sepuluh aturan emas yang membantu mengamankan sistem kontrol industri (ICS) Anda, Internet of Things (IIoT) industri, dan lingkungan cloud.
- [Praktik Terbaik Keamanan untuk PL Manufaktur](#) — Whitepaper ini menjelaskan praktik terbaik keamanan untuk merancang, menerapkan, dan merancang beban kerja manufaktur hybrid lokal ini untuk Cloud. AWS

Ketahanan di AWS IoT SiteWise

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

AWS IoT SiteWise sepenuhnya dikelola dan menggunakan AWS layanan yang sangat tersedia dan tahan lama, seperti Amazon S3 dan Amazon. EC2 Untuk memastikan ketersediaan jika terjadi gangguan zona ketersediaan, AWS IoT SiteWise beroperasi di beberapa zona ketersediaan.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Selain infrastruktur AWS global, AWS IoT SiteWise menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda:

- Anda dapat mempublikasikan pembaruan nilai properti AWS IoT Core melalui pesan MQTT, lalu mengonfigurasi aturan untuk menindaklanjuti data tersebut. Dengan fitur ini, Anda dapat mencadangkan data di AWS layanan lain seperti Amazon S3 dan Amazon DynamoDB. Untuk informasi selengkapnya, lihat [Berinteraksi dengan AWS layanan lain](#) dan [Ekspor data ke Amazon S3 dengan pemberitahuan properti aset](#).
- Anda dapat menggunakan AWS IoT SiteWise Get * APIs untuk mengambil dan membuat cadangan data properti aset historis. Untuk informasi selengkapnya, lihat [Kueri nilai properti aset historis di AWS IoT SiteWise](#).
- Anda dapat menggunakan AWS IoT SiteWise Describe* APIs untuk mengambil definisi untuk sumber daya Anda, seperti aset dan model. Anda dapat membuat cadangan definisi ini dan kemudian menggunakannya untuk membuat ulang sumber daya Anda. Untuk informasi lebih lanjut, lihat [Referensi API AWS IoT SiteWise](#).

Keamanan infrastruktur di AWS IoT SiteWise

Sebagai layanan terkelola, AWS IoT SiteWise dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS IoT SiteWise melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan [AWS Security Token](#)

[Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

SiteWise Edge gateway, yang berjalan AWS IoT Greengrass, menggunakan sertifikat X.509 dan kunci kriptografi untuk menghubungkan dan mengautentikasi ke Cloud. AWS Untuk informasi selengkapnya, lihat [Autentikasi dan otorisasi perangkat AWS IoT Greengrass](#) di Panduan AWS IoT Greengrass Version 1 Pengembang.

Analisis konfigurasi dan kerentanan di AWS IoT SiteWise

Armada IoT dapat terdiri dari sejumlah besar perangkat yang memiliki kemampuan beragam, beragam umur, dan didistribusikan secara geografis. Karakteristik ini membuat pengaturan armada kompleks dan rawan kesalahan. Karena perangkat biasanya memiliki daya pemrosesan, memori, dan penyimpanan yang terbatas, mereka tidak selalu dapat mendukung enkripsi dan langkah-langkah keamanan lainnya. Selain itu, perangkat sering menggunakan perangkat lunak dengan kerentanan yang diketahui. Faktor-faktor ini membuat armada IoT menjadi target yang menarik bagi peretas dan menyulitkan untuk mengamankan armada perangkat Anda secara berkelanjutan.

AWS IoT Device Defender mengatasi tantangan ini dengan menyediakan alat untuk mengidentifikasi masalah keamanan dan penyimpangan dari praktik terbaik. Gunakan AWS IoT Device Defender untuk menganalisis, mengaudit, dan memantau perangkat yang terhubung untuk mendeteksi perilaku abnormal, dan mengurangi risiko keamanan. AWS IoT Device Defender dapat mengaudit armada perangkat untuk memastikan mereka mematuhi praktik terbaik keamanan dan mendeteksi perilaku abnormal pada perangkat. Hal ini memungkinkan untuk menerapkan kebijakan keamanan yang konsisten di seluruh armada AWS IoT perangkat Anda dan merespons dengan cepat ketika perangkat dikompromikan. Untuk informasi selengkapnya, lihat [Apa yang ada AWS IoT Device Defender](#) di Panduan AWS IoT Device Defender Pengembang.

Jika Anda menggunakan gateway SiteWise Edge untuk menyerap data ke layanan, Anda bertanggung jawab untuk mengonfigurasi dan memelihara lingkungan gateway SiteWise Edge Anda. Tanggung jawab ini mencakup peningkatan ke versi terbaru dari perangkat lunak sistem, perangkat AWS IoT Greengrass lunak, dan AWS IoT SiteWise konektor SiteWise Edge gateway. Untuk informasi selengkapnya, lihat [Mengkonfigurasi AWS IoT Greengrass inti](#) di Panduan AWS IoT Greengrass Version 1 Pengembang dan [Kelola SiteWise gateway Edge](#).

Titik akhir VPC untuk AWS IoT SiteWise

Endpoint VPC antarmuka membuat koneksi pribadi antara virtual private cloud (VPC) Anda dan AWS IoT SiteWise [AWS PrivateLink](#) memberdayakan titik akhir antarmuka, memungkinkan akses pribadi ke operasi AWS IoT SiteWise API. Anda dapat melewati kebutuhan akan gateway internet, perangkat NAT, koneksi VPN, atau AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi AWS IoT SiteWise dengan operasi API. Lalu lintas antara VPC Anda dan AWS IoT SiteWise tidak meninggalkan jaringan. AWS

Setiap titik akhir antarmuka diwakili oleh satu atau lebih [antarmuka jaringan elastis](#) dalam subnet Anda.

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk AWS IoT SiteWise, tinjau [Akses AWS layanan menggunakan titik akhir VPC antarmuka](#) di Panduan.AWS PrivateLink

Operasi API untuk titik akhir VPC di AWS IoT SiteWise

AWS IoT SiteWise mendukung panggilan ke operasi AWS IoT SiteWise API berikut dari VPC Anda:

- Untuk semua operasi API bidang data, gunakan titik akhir berikut: Ganti *region* dengan Wilayah AWS

```
data.iotsitewise.region.amazonaws.com
```

Operasi API bidang data meliputi:

- [BatchGetAssetPropertyValue](#)
- [BatchGetAssetPropertyValueHistory](#)
- [BatchPutAssetPropertyValue](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [GetInterpolatedAssetPropertyValues](#)
- Untuk operasi API bidang kontrol yang Anda gunakan untuk mengelola model aset, aset, gateway SiteWise Edge, tag, dan konfigurasi akun, gunakan titik akhir berikut. Ganti *region* dengan Anda Wilayah AWS.

```
api.iotsitewise.region.amazonaws.com
```

Operasi API bidang kontrol yang didukung mencakup hal-hal berikut:

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAssetModel](#)
- [DeleteAsset](#)
- [DeleteAssetModel](#)
- [DeleteDashboard](#)
- [DescribeAsset](#)
- [DescribeAssetModel](#)
- [DescribeAssetProperty](#)
- [DescribeDashboard](#)
- [DescribeLoggingOptions](#)
- [DisassociateAssets](#)
- [ListAssetModels](#)
- [ListAssetRelationships](#)
- [ListAssets](#)
- [ListAssociatedAssets](#)
- [PutLoggingOptions](#)
- [UpdateAsset](#)
- [UpdateAssetModel](#)
- [UpdateAssetProperty](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorageConfiguration](#)
- [ListGateways](#)

- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorageConfiguration](#)
- [TagResource](#)
- [UntagResource](#)

 Note

Titik akhir VPC antarmuka untuk operasi API bidang kontrol saat ini tidak mendukung panggilan ke operasi SiteWise Monitor API berikut:

- [BatchAssociateProjectAssets](#)
- [BatchDisassociateProjectAssets](#)
- [CreateAccessPolicy](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccessPolicy](#)
- [DeletePortal](#)
- [DeleteProject](#)
- [DescribeAccessPolicy](#)
- [DescribePortal](#)
- [DescribeProject](#)
- [ListAccessPolicies](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProjectAssets](#)
- [UpdateAccessPolicy](#)

- [UpdatePortal](#)
- [UpdateProject](#)

Buat titik akhir VPC antarmuka untuk AWS IoT SiteWise

Untuk membuat titik akhir VPC untuk AWS IoT SiteWise layanan, gunakan konsol VPC Amazon atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat [Mengakses AWS layanan menggunakan titik akhir VPC antarmuka di Panduan](#). AWS PrivateLink

Buat titik akhir VPC AWS IoT SiteWise dengan menggunakan salah satu nama layanan berikut:

- Untuk operasi API bidang data, gunakan nama layanan berikut:

```
com.amazonaws.region.iotsitewise.data
```

- Untuk operasi API bidang kontrol, gunakan nama layanan berikut:

```
com.amazonaws.region.iotsitewise.api
```

Akses AWS IoT SiteWise melalui titik akhir VPC antarmuka

Saat Anda membuat titik akhir antarmuka, kami menghasilkan nama host DNS khusus titik akhir yang dapat Anda gunakan untuk berkomunikasi. AWS IoT SiteWise Opsi DNS pribadi diaktifkan secara default. Untuk informasi selengkapnya, lihat [Menggunakan zona yang dihosting pribadi](#) di Panduan Pengguna Amazon VPC.

Jika Anda mengaktifkan DNS pribadi untuk titik akhir, Anda dapat membuat permintaan API AWS IoT SiteWise melalui salah satu titik akhir VPC berikut.

- Untuk operasi API bidang data, gunakan titik akhir berikut: Ganti *region* dengan Wilayah AWS.

```
data.iotsitewise.region.amazonaws.com
```

- Untuk operasi API bidang kontrol, gunakan titik akhir berikut: Ganti *region* dengan Wilayah AWS.

```
api.iotsitewise.region.amazonaws.com
```

Jika Anda menonaktifkan DNS pribadi untuk titik akhir, Anda harus melakukan hal berikut untuk mengakses AWS IoT SiteWise melalui titik akhir:

1. Tentukan url titik akhir VPC dalam permintaan API.

- Untuk operasi API bidang data, gunakan url titik akhir berikut. Ganti *vpc-endpoint-id* dan *region* dengan ID titik akhir VPC dan Wilayah Anda.

```
vpc-endpoint-id.data.iotsitewise.region.vpce.amazonaws.com
```

- Untuk operasi API bidang kontrol, gunakan url titik akhir berikut. Ganti *vpc-endpoint-id* dan *region* dengan ID titik akhir VPC dan Wilayah Anda.

```
vpc-endpoint-id.api.iotsitewise.region.vpce.amazonaws.com
```

2. Nonaktifkan injeksi awalan host. AWS CLI Dan tambahkan AWS SDKs titik akhir layanan dengan berbagai awalan host saat Anda memanggil setiap operasi API. Fitur ini AWS SDKs menyebabkan AWS CLI dan menghasilkan URLs yang tidak valid AWS IoT SiteWise saat Anda menentukan titik akhir VPC.

 Important

Anda tidak dapat menonaktifkan injeksi awalan host di AWS CLI atau file. Alat AWS untuk PowerShell Ini berarti bahwa jika Anda menonaktifkan DNS pribadi, maka Anda tidak dapat menggunakan alat ini untuk mengakses AWS IoT SiteWise melalui titik akhir VPC. Aktifkan DNS pribadi untuk menggunakan AWS CLI atau Alat AWS untuk PowerShell untuk mengakses AWS IoT SiteWise melalui titik akhir.

Untuk informasi selengkapnya tentang cara menonaktifkan injeksi awalan host di AWS SDKs, lihat bagian dokumentasi berikut untuk setiap SDK:

- [AWS SDK untuk C++](#)
- [AWS SDK untuk Go](#)
- [AWS SDK untuk Go v2](#)
- [AWS SDK untuk Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK untuk JavaScript](#)

- [AWS SDK untuk .NET](#)
- [AWS SDK untuk PHP](#)
- [AWS SDK untuk Python \(Boto3\)](#)
- [AWS SDK untuk Ruby](#)

Untuk informasi selengkapnya, lihat [Mengakses AWS layanan menggunakan titik akhir VPC antarmuka di Panduan.AWS PrivateLink](#)

Membuat kebijakan titik akhir VPC untuk AWS IoT SiteWise

Anda dapat melampirkan kebijakan titik akhir ke VPC endpoint yang mengendalikan akses ke AWS IoT SiteWise. Kebijakan menentukan informasi berikut ini:

- Kepala sekolah yang dapat melakukan operasi.
- Operasi yang dapat dilakukan.
- Sumber daya di mana operasi dapat dilakukan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir di Panduan Pengguna Amazon VPC](#).

Contoh: Kebijakan titik akhir VPC untuk tindakan AWS IoT SiteWise

Berikut ini adalah contoh kebijakan endpoint untuk AWS IoT SiteWise. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke AWS IoT SiteWise tindakan yang tercantum untuk pengguna *iotsitewiseadmin* di AWS akun *123456789012* pada aset yang ditentukan.

```
{
  "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"  
  ]  
}  
]  
}
```

Praktik terbaik keamanan untuk AWS IoT SiteWise

Topik ini berisi praktik terbaik keamanan untuk AWS IoT SiteWise.

Gunakan kredensi otentikasi di server OPC UA Anda

Memerlukan kredensi otentikasi untuk terhubung ke server OPC UA Anda. Konsultasikan dokumentasi untuk server Anda untuk melakukannya. Kemudian, untuk memungkinkan gateway SiteWise Edge Anda terhubung ke server OPC UA Anda, tambahkan rahasia otentikasi server ke gateway SiteWise Edge Anda. Untuk informasi selengkapnya, lihat [Konfigurasi otentikasi sumber data untuk Edge SiteWise](#).

Gunakan mode komunikasi terenkripsi untuk server OPC UA Anda

Pilih mode keamanan pesan terenkripsi yang tidak digunakan lagi saat Anda mengonfigurasi sumber OPC UA untuk gateway Edge Anda. SiteWise Ini membantu mengamankan data industri Anda saat bergerak dari server OPC UA Anda ke gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Data dalam transit melalui jaringan lokal](#) dan [Siapkan sumber OPC UA di SiteWise Edge](#).

Perbarui komponen Anda

Jika Anda menggunakan gateway SiteWise Edge untuk menyerap data ke layanan, Anda bertanggung jawab untuk mengonfigurasi dan memelihara lingkungan gateway Edge Anda SiteWise. Tanggung jawab ini mencakup peningkatan ke versi terbaru dari perangkat lunak, perangkat AWS IoT Greengrass lunak, dan konektor sistem gateway.

Note

Konektor AWS IoT SiteWise Edge menyimpan rahasia di sistem file Anda. Rahasia ini mengontrol siapa yang dapat melihat data yang di-cache dalam gateway SiteWise Edge Anda. Sangat disarankan agar Anda mengaktifkan enkripsi disk atau file-system untuk sistem yang menjalankan gateway SiteWise Edge Anda.

Untuk informasi tentang cara memutakhirkan komponen di AWS IoT SiteWise konsol, lihat [Ubah versi paket komponen gateway SiteWise Edge](#).

Enkripsi sistem file gateway SiteWise Edge Anda

Enkripsi dan amankan gateway SiteWise Edge Anda, sehingga data industri Anda aman saat bergerak melalui gateway SiteWise Edge. Jika gateway SiteWise Edge Anda memiliki modul keamanan perangkat keras, Anda dapat mengonfigurasi AWS IoT Greengrass untuk mengamankan gateway SiteWise Edge Anda. Untuk informasi selengkapnya, lihat [Integrasi keamanan perangkat keras](#) di Panduan AWS IoT Greengrass Version 1 Pengembang. Jika tidak, lihat dokumentasi untuk sistem operasi Anda untuk mempelajari cara mengenkripsi dan mengamankan sistem file Anda.

Akses aman ke konfigurasi tepi Anda

Jangan bagikan kata sandi aplikasi konsol tepi atau kata sandi aplikasi SiteWise Monitor Anda. Jangan letakkan kata sandi ini di tempat di mana siapa pun dapat melihatnya. Terapkan kebijakan rotasi kata sandi yang sehat dengan mengonfigurasi kedaluwarsa yang sesuai untuk kata sandi Anda.

Mengamankan data Siemens Industrial Edge Management

Data perangkat yang Anda pilih untuk dibagikan dengan AWS IoT SiteWise Edge ditentukan dalam topik Siemens IEM Databus konfigurasi Anda. Dengan memilih topik untuk dibagikan dengan SiteWise Edge, Anda berbagi data tingkat topik dengan AWS IoT SiteWise Siemens Industrial Edge Marketplace. Ini adalah pasar independen, terpisah dari AWS. Untuk melindungi data bersama Anda, aplikasi SiteWise Edge tidak akan berjalan kecuali Anda menggunakannya Siemens Secured Storage. Untuk informasi selengkapnya, lihat [Penyimpanan Aman](#), dalam Siemens dokumentasi.

Grant SiteWise Monitor pengguna izin minimum yang mungkin

Ikuti prinsip hak istimewa terkecil dengan menggunakan set minimum izin kebijakan akses untuk pengguna portal Anda.

- Saat Anda membuat portal, tentukan peran yang memungkinkan kumpulan aset minimum yang diperlukan untuk portal tersebut. Untuk informasi selengkapnya, lihat [Gunakan peran layanan untuk AWS IoT SiteWise Monitor](#).
- Saat Anda dan administrator portal membuat dan berbagi proyek, gunakan kumpulan aset minimum yang diperlukan untuk proyek tersebut.

- Ketika identitas tidak lagi membutuhkan akses ke portal atau proyek, hapus mereka dari sumber daya itu. Jika identitas itu tidak lagi berlaku untuk organisasi Anda, hapus identitas itu dari toko identitas Anda.

Praktik terbaik prinsip terkecil juga berlaku untuk peran IAM. Untuk informasi selengkapnya, lihat [Praktik terbaik kebijakan](#).

Jangan mengekspos informasi sensitif

Anda harus mencegah pencatatan kredensial dan informasi sensitif lainnya, seperti informasi identitas pribadi (PII). Kami menyarankan Anda menerapkan perlindungan berikut meskipun akses ke log lokal di gateway SiteWise Edge memerlukan hak akses root dan akses ke CloudWatch Log memerlukan izin IAM.

- Jangan gunakan informasi sensitif dalam nama, deskripsi, atau properti aset atau model Anda.
- Jangan gunakan informasi sensitif di gateway SiteWise Edge atau nama sumber.
- Jangan gunakan informasi sensitif dalam nama atau deskripsi portal, proyek, atau dasbor Anda.

Ikuti praktik terbaik AWS IoT Greengrass keamanan

Ikuti praktik terbaik AWS IoT Greengrass keamanan untuk gateway SiteWise Edge Anda. Untuk informasi selengkapnya, lihat [Praktik terbaik keamanan](#) di Panduan AWS IoT Greengrass Version 1 Pengembang.

Lihat juga

- [Praktik terbaik keamanan](#) di Panduan AWS IoT Pengembang
- [Sepuluh aturan emas keamanan untuk solusi IoT Industri](#)

Masuk dan pantau AWS IoT SiteWise

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja AWS IoT SiteWise dan AWS solusi Anda yang lain. AWS IoT SiteWise mendukung alat pemantauan berikut untuk menonton layanan, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Kumpulkan dan lacak metrik, buat dasbor khusus, dan setel alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas tertentu. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari EC2 instans Amazon Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).
- Amazon CloudWatch Logs memantau, menyimpan, dan mengakses file log Anda dari gateway SiteWise Edge CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Logs](#).
- AWS CloudTrail menangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda. Kemudian CloudTrail mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).

Topik

- [Monitor dengan Amazon CloudWatch Logs](#)
- [Pantau log gateway SiteWise Edge](#)
- [Pantau AWS IoT SiteWise dengan CloudWatch metrik Amazon](#)
- [Log panggilan AWS IoT SiteWise API dengan AWS CloudTrail](#)

Monitor dengan Amazon CloudWatch Logs

Konfigurasi AWS IoT SiteWise untuk mencatat informasi ke CloudWatch Log untuk memantau dan memecahkan masalah layanan.

Saat Anda menggunakan AWS IoT SiteWise konsol, AWS IoT SiteWise buat peran terkait layanan yang memungkinkan layanan mencatat informasi atas nama Anda. Jika Anda tidak menggunakan AWS IoT SiteWise konsol, Anda harus membuat peran terkait layanan secara manual untuk menerima log. Untuk informasi selengkapnya, lihat [Membuat peran terkait layanan untuk AWS IoT SiteWise](#).

Anda harus memiliki kebijakan sumber daya yang memungkinkan AWS IoT SiteWise untuk menempatkan peristiwa log ke CloudWatch aliran. Untuk membuat dan memperbarui kebijakan sumber daya untuk CloudWatch Log, jalankan perintah berikut. Ganti *logging-policy-name* dengan nama kebijakan yang akan dibuat.

```
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 Log juga mendukung kunci konteks SourceAccount kondisi [aws: SourceArn](#) dan [aws:](#). Kunci konteks kondisi ini bersifat opsional.

Untuk membuat atau memperbarui kebijakan sumber daya yang memungkinkan AWS IoT SiteWise untuk hanya menempatkan log yang terkait dengan AWS IoT SiteWise sumber daya tertentu ke dalam CloudWatch aliran, jalankan perintah dan lakukan hal berikut:

- Ganti *logging-policy-name* dengan nama kebijakan yang akan dibuat.
- Ganti *source-ARN* dengan ARN AWS IoT SiteWise sumber daya Anda, seperti model aset atau aset. Untuk menemukan ARN untuk setiap jenis sumber daya, lihat Jenis AWS IoT SiteWise sumber daya yang [ditentukan oleh AWS IoT SiteWise](#) dalam Referensi Otorisasi Layanan.
- Ganti *account-ID* dengan ID AWS akun yang terkait dengan AWS IoT SiteWise sumber daya yang ditentukan.

```
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\"] } } } ] }"
```

Secara default, AWS IoT SiteWise tidak mencatat informasi ke CloudWatch Log. Untuk mengaktifkan logging, pilih level logging selain Disabled (OFF). AWS IoT SiteWise mendukung tingkat logging berikut:

- OFF— Logging dimatikan.
- ERROR— Kesalahan dicatat.
- INFO— Kesalahan dan pesan informasi dicatat.

Anda dapat mengonfigurasi gateway SiteWise Edge untuk mencatat informasi ke CloudWatch Log melalui AWS IoT Greengrass. Untuk informasi selengkapnya, lihat [Pantau log gateway SiteWise Edge](#).

Anda juga dapat mengonfigurasi AWS IoT Core untuk mencatat informasi ke CloudWatch Log jika Anda memecahkan masalah tindakan AWS IoT SiteWise aturan. Untuk informasi selengkapnya, lihat [Memecahkan masalah tindakan aturan AWS IoT SiteWise](#).

Daftar Isi

- [Mengelola masuk AWS IoT SiteWise](#)
 - [Temukan tingkat pencatatan Anda](#)
 - [Ubah level logging Anda](#)
- [Contoh: entri file AWS IoT SiteWise log](#)

Mengelola masuk AWS IoT SiteWise

Gunakan AWS IoT SiteWise konsol atau AWS CLI untuk tugas konfigurasi logging berikut.

Temukan tingkat pencatatan Anda

Console

Gunakan prosedur berikut untuk menemukan level logging Anda saat ini di AWS IoT SiteWise konsol.

Untuk menemukan level AWS IoT SiteWise logging Anda saat ini

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Opsi Pencatatan.

Status logging saat ini muncul di bawah status Logging. Jika logging diaktifkan, level logging saat ini muncul di bawah Level verbositas.

AWS CLI

Jalankan perintah berikut untuk menemukan level AWS IoT SiteWise logging Anda saat ini dengan file AWS CLI.

```
aws iotsitewise describe-logging-options
```

Operasi mengembalikan respons yang berisi tingkat logging Anda dalam format berikut.

```
{
  "loggingOptions": {
    "level": "String"
  }
}
```

Ubah level logging Anda

Gunakan prosedur berikut untuk mengubah level logging Anda di AWS IoT SiteWise konsol atau menggunakan AWS CLI.

Console

Untuk mengubah level AWS IoT SiteWise logging Anda

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi kiri, pilih Opsi Pencatatan.
3. Pilih Edit.
4. Pilih Level verbositas untuk diaktifkan.
5. Pilih Simpan.

AWS CLI

Jalankan AWS CLI perintah berikut untuk mengubah level AWS IoT SiteWise logging Anda. Ganti *logging-level* dengan level logging yang Anda inginkan.

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

Contoh: entri file AWS IoT SiteWise log

Setiap entri AWS IoT SiteWise log mencakup informasi peristiwa dan sumber daya yang relevan untuk acara tersebut, sehingga Anda dapat memahami dan menganalisis data log.

Contoh berikut menunjukkan entri CloudWatch AWS IoT SiteWise Log yang mencatat saat Anda berhasil membuat model aset.

```
{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",
  "eventType": "AssetModelCreationSuccess",
  "message": "Successfully created asset model.",
  "resources": {
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
}
```

Pantau log gateway SiteWise Edge

Anda dapat mengonfigurasi gateway AWS IoT SiteWise Edge Anda untuk mencatat informasi ke Amazon CloudWatch Logs atau sistem file lokal.

Topik

- [Gunakan CloudWatch Log Amazon](#)
- [Gunakan log layanan di AWS IoT SiteWise](#)

Gunakan CloudWatch Log Amazon

Anda dapat mengonfigurasi gateway SiteWise Edge Anda untuk mengirim log ke CloudWatch Log. Untuk informasi selengkapnya, lihat [Mengaktifkan pencatatan untuk CloudWatch Log](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

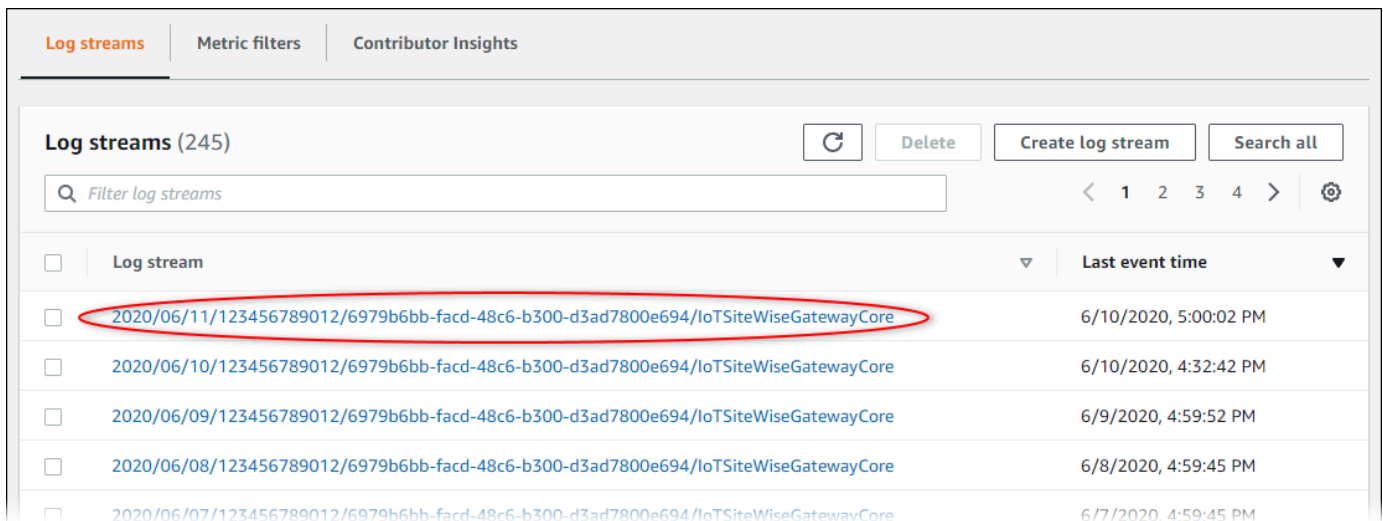
Untuk mengkonfigurasi dan mengakses CloudWatch Log (Konsol)

1. Navigasikan ke [konsol CloudWatch](#) tersebut.

2. Pada panel navigasi, pilih Grup log.
3. Anda dapat menemukan log AWS IoT SiteWise komponen di grup log berikut:
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollectorOpcua`— Log untuk komponen gateway SiteWise Edge yang mengumpulkan data dari sumber OPC UA gateway SiteWise Edge.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— Log untuk komponen gateway SiteWise Edge yang menerbitkan aliran data OPC UA ke. AWS IoT SiteWise

Pilih grup log untuk fungsi yang akan di-debug.

4. Pilih aliran log yang memiliki nama yang diakhiri dengan nama AWS IoT Greengrass grup Anda. Secara default, CloudWatch menampilkan aliran log terbaru terlebih dahulu.



5. Untuk menampilkan log dari 5 menit terakhir, lakukan hal berikut:
 - a. Pilih kustom di sudut kanan atas.
 - b. Pilih Relatif.
 - c. Pilih 5 menit.
 - d. Pilih Terapkan.

6. (Opsional) Untuk melihat lebih sedikit log, Anda dapat memilih 1m dari sudut kanan atas.
7. Gulir ke bagian bawah entri log untuk menampilkan log terbaru.

Gunakan log layanan di AWS IoT SiteWise

SiteWise Perangkat gateway tepi menyertakan file log layanan untuk membantu masalah debug. Bagian berikut akan membantu Anda menemukan dan memanfaatkan file log layanan untuk komponen AWS IoT SiteWise OPC UA Collector dan AWS IoT SiteWise Publisher.

AWS IoT SiteWise File log layanan OPC UA Collector

Komponen AWS IoT SiteWise OPC UA Collector menggunakan file log berikut.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Untuk melihat log komponen ini

- Jalankan perintah berikut pada perangkat inti untuk melihat file log komponen ini secara real time. Ganti `/greengrass/v2` atau `C:\greengrass\v2` dengan jalur ke folder AWS IoT Greengrass root.

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 File log layanan penerbit

Komponen AWS IoT SiteWise Publisher menggunakan file log berikut.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

Untuk melihat log komponen ini

- Jalankan perintah berikut pada perangkat inti untuk melihat file log komponen ini secara real time. Ganti `/greengrass/v2` atau `C:\greengrass\v2` dengan jalur ke folder AWS IoT Greengrass root.

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
```

Pantau AWS IoT SiteWise dengan CloudWatch metrik Amazon

Anda dapat memantau AWS IoT SiteWise penggunaan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

AWS IoT SiteWise menerbitkan metrik dan dimensi yang tercantum di bagian di bawah ini ke namespace. AWS/IoTSiteWise

Tip

AWS IoT SiteWise menerbitkan metrik pada interval satu menit. Saat Anda melihat metrik ini dalam grafik di CloudWatch konsol, kami sarankan Anda memilih Periode 1 menit. Ini memungkinkan Anda melihat resolusi tertinggi yang tersedia dari data metrik Anda.

Topik

- [AWS IoT Greengrass Version 2 metrik gateway](#)

AWS IoT Greengrass Version 2 metrik gateway

AWS IoT SiteWise menerbitkan metrik gateway untuk aliran Klasik, gateway V2, dan gateway V3 berkemampuan MQTT. Kecuali dinyatakan lain, setiap metrik berlaku untuk kedua versi gateway yang dihosting sendiri. Semua metrik gateway SiteWise Edge dipublikasikan dalam interval satu menit.

SiteWise Metrik gerbang tepi

Metrik	Deskripsi
<code>Gateway.AvailableMemory</code>	Memori yang tersedia dari gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada
<code>Gateway.AvailableDiskSpace</code>	Ruang disk yang tersedia dari gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada
<code>Gateway.CloudConnectivity</code>	Status konektivitas cloud dari gateway SiteWise Edge. Satuan: Tidak ada Dimensi: GatewayId
<code>Gateway.CpuUsage</code>	Penggunaan CPU dari gateway SiteWise Edge. Unit: Persentase Dimensi: Tidak ada
<code>Gateway.TotalDiskSpace</code>	Total ruang disk gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada
<code>Gateway.TotalMemory</code>	Memori total gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada

Metrik	Deskripsi
<code>Gateway.UsedDiskSpace</code>	Ruang disk yang digunakan dari gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada
<code>Gateway.UsedMemory</code>	Memori yang digunakan dari gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada
<code>Gateway.UsedPercentageDiskSpace</code>	Persentase ruang disk yang digunakan dari gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada
<code>Gateway.UsedPercentageMemory</code>	Memori persentase yang digunakan dari gateway SiteWise Edge. Unit: Bit Dimensi: Tidak ada

AWS IoT SiteWise metrik penerbit

Metrik	Deskripsi
<code>IoTSiteWisePublisher.ComponentBuildVersion</code>	Metrik ini menunjukkan versi build komponen SiteWise penerbit IoT yang berjalan di gateway. Nilai 1 menandakan bahwa gateway menjalankan versi penerbit yang sesuai dengan dimensi. <code>ComponentBuildVersion</code>

Metrik	Deskripsi
	Satuan: 1 Dimensi: GatewayId, ComponentBuildVersion
<code>IoTSiteWisePublisher.DroppedCount</code>	Jumlah titik data yang dijatuhkan oleh gateway SiteWise Edge (GatewayId) dan tidak dipublikasikan ke cloud, dihasilkan setiap menit. Unit: Jumlah Dimensi: GatewayId
<code>IoTSiteWisePublisher.Heartbeat</code>	Dihasilkan setiap menit oleh Penerbit di gateway SiteWise Edge. Unit: 1 (1 mewakili Penerbit sedang berjalan dan kehilangan titik data yang mewakili Penerbit tidak berjalan.) Dimensi: GatewayId
<code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code>	Dihasilkan setiap menit oleh Penerbit di gateway SiteWise Edge. Unit: 1 (1 mewakili Penerbit terhubung ke broker MQTT.) Dimensi: GatewayId
<code>IoTSiteWisePublisher.MessageCheckpointPersistenceErrorCount</code>	Metrik menunjukkan bahwa gateway telah mendeteksi masalah dengan file pos pemeriksaan yang digunakan untuk melacak data yang diproses oleh penerbit. Nilai 1 menandakan bahwa kegagalan telah terjadi. Satuan: Tidak ada Dimensi: AccountId, GatewayId

Metrik	Deskripsi
<code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code>	Jumlah pesan yang berhasil diterima oleh Penerbit dari broker MQTT, dihasilkan setiap menit. Unit: Jumlah Dimensi: GatewayId
<code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code>	Jumlah byte data pesan yang berhasil diterima oleh Penerbit dari broker MQTT, dihasilkan setiap menit. Unit: Jumlah Dimensi: GatewayId
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	Jumlah topik yang berlangganan broker MQTT oleh Penerbit, dihasilkan setiap menit. Topik kartu liar bertingkat dihitung sebagai 1. Unit: Jumlah Dimensi: GatewayId
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	Jumlah topik unik yang diterima oleh Penerbit dari broker MQTT, dihasilkan setiap menit. Unit: Jumlah Dimensi: GatewayId
<code>IoTSiteWisePublisher.PublishFailureCount</code>	Jumlah titik data yang gagal dipublikasikan oleh gateway SiteWise Edge (GatewayId), dihasilkan setiap menit. Unit: Jumlah Dimensi: GatewayId

Metrik	Deskripsi
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	Jumlah titik data yang ditolak oleh gateway SiteWise Edge (<code>GatewayId</code>) dari sisi cloud, dihasilkan setiap menit. Unit: Jumlah Dimensi: <code>GatewayId</code>
<code>IoTSiteWisePublisher.PublisherSuccessCount</code>	Jumlah titik data yang berhasil dipublikasikan oleh gateway SiteWise Edge (<code>GatewayId</code>) ke cloud, dihasilkan setiap menit. Unit: Jumlah Dimensi: <code>GatewayId</code>
<code>IoTSiteWisePublisher.PublisherToS3FailureCount</code>	Jumlah titik data yang gagal dipublikasikan oleh gateway (<code>GatewayId</code>) ke bucket Amazon S3. Unit: Jumlah Dimensi: <code>GatewayId</code>
<code>IoTSiteWisePublisher.PublisherToS3SuccessCount</code>	Jumlah titik data yang berhasil dipublikasikan gateway (<code>GatewayId</code>) ke bucket Amazon S3. Unit: Jumlah Dimensi: <code>GatewayId</code>

Metrik kolektor OPC UA

Metrik	Deskripsi
<code>OpCuaCollector.ActiveDataStreamCount</code>	Jumlah aliran data yang dilanggan gateway SiteWise Edge (<code>gatewayId</code>) untuk sumber OPC UA (<code>sourceName</code>).

Metrik	Deskripsi
	Unit: Jumlah Dimensi: GatewayId, SourceName, PropertyGroup
<code>OpcUaCollector.ComponentBuildVersion</code> (tidak tersedia di aliran Klasik, gateway V2)	Metrik ini menunjukkan versi build komponen kolektor IoT SiteWise OPC UA yang berjalan di gateway. Nilai 1 menandakan bahwa gateway menjalankan versi kolektor yang sesuai dengan <code>ComponentBuildVersion</code> dimensi. Satuan: 1 Dimensi: GatewayId, ComponentBuildVersion
<code>OpcUaCollector.ConversionErrors</code>	Jumlah titik data yang diterima gateway SiteWise Edge (gatewayId) untuk sumber OPC UA (sourceName) yang mengakibatkan kesalahan konversi saat mengirim data ke AWS IoT SiteWise. Titik data ini tidak akan dicerna oleh OPC UA Collector. Unit: Jumlah Dimensi: GatewayId, SourceName
<code>OpcUaCollector.Heartbeat</code>	Dihasilkan setiap menit untuk setiap sumber OPC UA (sourceName) yang terhubung ke gateway SiteWise Edge (gatewayId). Satuan: Hitung (1 mewakili sumber terhubung dan 0 mewakili sumber terputus.) Dimensi: GatewayId, SourceName

Metrik	Deskripsi
<code>OpcUaCollector.IncomingValuesCount</code>	Jumlah titik data yang diterima gateway SiteWise Edge (<code>gatewayId</code>) untuk sumber OPC UA (<code>sourceName</code>), dihasilkan setiap menit. Unit: Jumlah Dimensi: <code>GatewayId</code>, <code>SourceName</code>, <code>PropertyGroup</code>
<code>OpcUaCollector.IncomingValueErrors</code>	Jumlah titik data yang diterima gateway SiteWise Edge (<code>gatewayId</code>) dari sumber OPC UA (<code>sourceName</code>) yang bukan nilai yang valid. Titik data ini tidak dicerna oleh OPC UA Collector, yang dihasilkan setiap menit. Unit: Jumlah Dimensi: <code>GatewayId</code>, <code>SourceName</code>, <code>PropertyGroup</code>
<code>OpcUaCollector.IsConnectedToMqttBroker</code> (tidak tersedia di aliran Klasik, gateway V2)	Dihasilkan setiap menit oleh komponen kolektor IoT SiteWise OPC UA di gateway Edge. SiteWise Unit: 1 (1 mewakili komponen kolektor IoT SiteWise OPC UA terhubung ke broker MQTT) Dimensi: <code>GatewayId</code>
<code>OpcUaCollector.MqttMessagesDroppedCount</code> (tidak tersedia di aliran Klasik, gateway V2)	Jumlah pesan MQTT yang dijatuhkan oleh komponen kolektor IoT OPC UA SiteWise . Unit: Jumlah Dimensi: <code>GatewayId</code>, <code>SourceName</code>

Metrik	Deskripsi
<code>OpcUaCollector.MqttMessagesPublishedBytes</code> (tidak tersedia di aliran Klasik, gateway V2)	Jumlah byte data pesan MQTT yang berhasil diterbitkan oleh komponen kolektor IoT SiteWise OPC UA ke broker MQTT. Unit: Jumlah Dimensi: GatewayId, SourceName
<code>OpcUaCollector.MqttMessagesPublishedCount</code> (tidak tersedia di aliran Klasik, gateway V2)	Jumlah pesan MQTT yang berhasil diterbitkan oleh komponen kolektor IoT SiteWise OPC UA ke broker MQTT. Unit: Jumlah Dimensi: GatewayId, SourceName
<code>OpcUaCollector.NullValueCount</code> (tidak tersedia di aliran Klasik, gateway V2)	Jumlah nilai nol yang diterima oleh komponen kolektor SiteWise IoT OPC UA dari server OPC UA. Unit: Jumlah Dimensi: GatewayId, SourceName, PropertyGroup
<code>OpcUaCollector.NumberOfUniqueMqttTopicsPublished</code> (tidak tersedia di aliran Klasik, gateway V2)	Jumlah topik MQTT unik yang diterbitkan oleh kolektor IoT SiteWise OPC UA ke broker MQTT. Unit: Jumlah Dimensi: GatewayId, SourceName

AWS IoT SiteWise metrik prosesor

Metrik	Deskripsi
<code>Gateway.DataProcessor.IngestionThrottled</code> (tidak tersedia di gateway V3 berkemampuan MQTT)	Jumlah titik data yang dibatasi, dihasilkan setiap menit. Unit: Jumlah Dimensi: <code>ThrottledAt</code>
<code>Gateway.DataProcessor.MeasurementRejected</code> (tidak tersedia di gateway V3 berkemampuan MQTT)	Jumlah pengukuran yang ditolak, dihasilkan setiap menit. Unit: Jumlah Dimensi: Alasan
<code>Gateway.DataProcessor.MessagesRemaining</code> (tidak tersedia di gateway V3 berkemampuan MQTT)	Jumlah pesan yang tersisa dalam aliran, dihasilkan setiap menit. Unit: Jumlah Dimensi: <code>StreamName</code>
<code>Gateway.DataProcessor.ProcessingError</code> (tidak tersedia di gateway V3 berkemampuan MQTT)	Jumlah kesalahan pemrosesan, dihasilkan setiap menit. Unit: Jumlah Dimensi: Alasan

Log panggilan AWS IoT SiteWise API dengan AWS CloudTrail

AWS IoT SiteWise terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS IoT SiteWise. CloudTrail menangkap panggilan API untuk AWS IoT SiteWise sebagai acara. Panggilan yang diambil termasuk panggilan dari AWS IoT SiteWise konsol dan panggilan kode ke operasi AWS IoT SiteWise API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk. AWS IoT SiteWise Jika Anda tidak mengonfigurasi jejak,

Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat AWS IoT SiteWise, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

AWS IoT SiteWise informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas acara yang didukung terjadi di AWS IoT SiteWise, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan riwayat CloudTrail acara](#).

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk AWS IoT SiteWise, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail layanan dan integrasi yang didukung](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa Wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan itu dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#).

AWS IoT SiteWise peristiwa data di CloudTrail

[Peristiwa data](#) memberikan informasi tentang operasi sumber daya yang dilakukan pada atau di sumber daya (misalnya, membaca atau menulis ke objek Amazon S3). Ini juga dikenal sebagai operasi bidang data. Peristiwa data seringkali merupakan aktivitas volume tinggi. Secara default, CloudTrail tidak mencatat peristiwa data. Riwayat CloudTrail peristiwa tidak merekam peristiwa data.

Biaya tambahan berlaku untuk peristiwa data. Untuk informasi selengkapnya tentang CloudTrail harga, lihat [AWS CloudTrail Harga](#).

Anda dapat mencatat peristiwa data untuk jenis AWS IoT SiteWise sumber daya menggunakan CloudTrail konsol AWS CLI, atau operasi CloudTrail API. [Tabel](#) di bagian ini menunjukkan jenis sumber daya yang tersedia untuk AWS IoT SiteWise.

- Untuk mencatat peristiwa data menggunakan CloudTrail konsol, buat [penyimpanan data jejak atau peristiwa](#) untuk mencatat peristiwa data, atau [perbarui penyimpanan data jejak atau peristiwa yang ada](#) untuk mencatat peristiwa data.
 1. Pilih Peristiwa data untuk mencatat peristiwa data.
 2. Dari daftar tipe peristiwa Data, pilih jenis sumber daya yang ingin Anda log peristiwa data.
 3. Pilih template pemilih log yang ingin Anda gunakan. Anda dapat mencatat semua peristiwa data untuk jenis sumber daya, mencatat semua `readOnly` peristiwa, mencatat semua `writeOnly` peristiwa, atau membuat templat pemilih log kustom untuk memfilter pada `readOnlyeventName`, dan `resources.ARN` bidang.
- Untuk mencatat peristiwa data menggunakan AWS CLI, konfigurasikan `--advanced-event-selectors` parameter untuk mengatur `eventCategory` bidang sama dengan Data dan `resources.type` bidang sama dengan nilai tipe sumber daya (lihat [tabel](#)). Anda dapat menambahkan kondisi untuk memfilter nilai `readOnly`, `eventName`, dan `resources.ARN` bidang.
- Untuk mengonfigurasi jejak untuk mencatat peristiwa data, jalankan [AWS CloudTrail put-event-selectors](#) perintah. Untuk informasi selengkapnya, lihat [Mencatat peristiwa data untuk jejak dengan AWS CLI](#)
- Untuk mengonfigurasi penyimpanan data peristiwa untuk mencatat peristiwa data, jalankan [AWS CloudTrail create-event-data-store](#) perintah untuk membuat penyimpanan data peristiwa baru untuk mencatat peristiwa data, atau jalankan [AWS CloudTrail update-event-data-store](#) perintah untuk memperbarui penyimpanan data peristiwa yang ada. Untuk informasi selengkapnya, lihat [Mencatat peristiwa data untuk menyimpan data peristiwa dengan AWS CLI](#).

Tabel berikut mencantumkan jenis AWS IoT SiteWise sumber daya. Kolom tipe peristiwa data (konsol) menunjukkan nilai yang akan dipilih dari daftar tipe peristiwa Data di CloudTrail konsol. Kolom nilai `resources.type` menunjukkan **resources.type** nilai, yang akan Anda tentukan saat mengonfigurasi penyeleksi acara lanjutan menggunakan `or`. AWS CLI CloudTrail APIs CloudTrailKolom Data yang APIs dicatat ke menampilkan panggilan API yang dicatat CloudTrail untuk jenis sumber daya.

Jenis peristiwa data (konsol)	nilai <code>resources.type</code>	Data APIs masuk CloudTrail ke*
AWS IoT SiteWise aset	<code>AWS::IoTSiteWise::Asset</code>	• BatchPutAssetPropertyValues • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValues • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise deret waktu	<code>AWS::IoTSiteWise::TimeSeries</code>	• BatchPutAssetPropertyValues • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk CloudTrail ke*
		• BatchGetAssetPropertyValues • BatchGetAssetPropertyValuesHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise Asisten	AWS::SitewiseAssistant::Conversation	• InvokeAssistant

Note

Resources.type yang dicatat dalam peristiwa Cloudtrail bergantung pada pengenalan yang digunakan dalam permintaan API. Jika id aset ditentukan dalam permintaan maka Asset resources.type dicatat, jika tidak resources.type dicatat. TimeSeries

* Anda dapat mengonfigurasi pemilih acara lanjutan untuk memfilter pada `eventName`, `readOnly`, dan `resources.ARN` bidang untuk mencatat hanya peristiwa yang penting bagi Anda. Untuk informasi selengkapnya tentang bidang ini, lihat [AdvancedFieldSelector](#).

AWS IoT SiteWise acara manajemen di CloudTrail

[Peristiwa manajemen logging](#) memberikan informasi tentang operasi manajemen yang dilakukan pada sumber daya di AWS akun Anda. Ini juga dikenal sebagai operasi pesawat kontrol. Secara default, CloudTrail mencatat peristiwa manajemen.

AWS IoT SiteWise mencatat semua operasi pesawat AWS IoT SiteWise kontrol sebagai peristiwa manajemen. Untuk daftar operasi bidang AWS IoT SiteWise kontrol yang AWS IoT SiteWise masuk ke log CloudTrail, lihat [Referensi AWS IoT SiteWise API](#).

Contoh: entri file AWS IoT SiteWise log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili

permintaan tunggal dari sumber mana pun dan mencakup informasi tentang operasi yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateAsset operasi.

```
{
  "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-bbbbbbEXAMPLE",  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "123456789012"  
}
```

Tandai AWS IoT SiteWise sumber daya Anda

Menandai AWS IoT SiteWise sumber daya Anda menyediakan cara yang ampuh untuk mengkategorikan, mengelola, dan mengambil aset organisasi secara efisien. Dengan menetapkan tag, yang terdiri dari pasangan nilai kunci, Anda dapat melampirkan metadata deskriptif ke sumber daya Anda. Metadata dari tag dapat digunakan untuk merampingkan operasi. Misalnya, dalam skenario ladang angin, tag memungkinkan Anda memberi label turbin dengan atribut tertentu seperti lokasi, kapasitas, dan status operasional, memungkinkan identifikasi dan manajemen cepat di dalamnya. AWS IoT SiteWise

Mengintegrasikan tag dengan kebijakan AWS Identity and Access Management (IAM) meningkatkan keamanan dan kontrol operasional dengan mendefinisikan aturan akses bersyarat. Ini berarti Anda dapat menentukan bahwa hanya pengguna dengan tag tertentu. Misalnya, hanya mereka yang ditandai dengan peran atau departemen tertentu, yang dapat mengakses atau memodifikasi sumber daya tertentu.

Gunakan tag di AWS IoT SiteWise

Gunakan tag untuk mengkategorikan AWS IoT SiteWise sumber daya Anda berdasarkan tujuan, pemilik, lingkungan, atau klasifikasi lainnya untuk kasus penggunaan Anda. Ketika Anda memiliki banyak sumber daya dari jenis yang sama, Anda dapat dengan cepat mengidentifikasi sumber daya tertentu berdasarkan tag-nya.

Setiap tag terdiri dari kunci dan nilai opsional yang Anda tentukan. Misalnya, Anda dapat membuat serangkaian tag untuk model aset Anda untuk melacaknya sesuai dengan proses industri yang mereka dukung. Disarankan untuk mengembangkan satu set kunci tag yang disesuaikan untuk setiap jenis sumber daya yang Anda kelola. Menggunakan satu set kunci tag yang konsisten dapat membuatnya lebih mudah mengelola sumber daya.

Tag dengan AWS Management Console

Editor Tag di AWS Management Console menyediakan cara terpusat dan terpadu bagi Anda untuk membuat dan mengelola tag Anda untuk sumber daya dari semua AWS layanan. Untuk informasi selengkapnya, lihat [Memulai Editor Tag](#) di AWS Sumber Daya Penandaan dan Panduan Pengguna Editor Tag.

Tag dengan AWS IoT SiteWise API

AWS IoT SiteWise API juga menggunakan tag. Sebelum Anda membuat tag, perhatikan pembatasan penandaan. Untuk informasi selengkapnya, lihat [Konvensi penamaan dan penggunaan tag](#) di.

Referensi Umum AWS

- Untuk menambahkan tanda selama pembuatan sumber daya, tentukan tag dalam properti `tags` sumber daya.
- Untuk menambahkan tag ke sumber daya yang ada, atau untuk memperbarui nilai tag, gunakan [TagResource](#) operasi.
- Untuk menghapus tag dari sumber daya, gunakan [UntagResource](#) operasi.
- Untuk mengambil tag yang terkait dengan sumber daya, gunakan [ListTagsForResource](#) operasi, atau jelaskan sumber daya dan periksa properti `tags`.

Tabel berikut mencantumkan sumber daya yang dapat Anda beri tag menggunakan AWS IoT SiteWise API `Create` dan terkait serta `Describe` operasinya.

Sumber daya yang dapat diberi tag AWS IoT SiteWise

Sumber Daya	Buat operasi	Jelaskan operasi
Model aset atau model komponen	CreateAssetModel	DescribeAssetModel
Aset	CreateAsset	DescribeAsset
SiteWise Gerbang tepi	CreateGateway	DescribeGateway
Portal	CreatePortal	DescribePortal
Proyek	CreateProject	DescribeProject
Dasbor	CreateDashboard	DescribeDashboard
Kebijakan akses	CreateAccessPolicy	DescribeAccessPolicy
Deret waktu	BatchPutAssetPropertyValue	DescribeTimeSeries

Untuk [BatchPutAssetPropertyValue](#), Anda dapat mengonfigurasi sumber data Anda untuk mengirim data industri AWS IoT SiteWise sebelum Anda membuat model aset dan aset. AWS IoT SiteWise secara otomatis membuat aliran data untuk menerima aliran data mentah dari peralatan Anda. Untuk informasi selengkapnya, lihat [Mengelola konsumsi data](#).

Gunakan operasi berikut untuk melihat dan mengelola tag untuk sumber daya yang mendukung penandaan:

- [TagResource](#)— Menambahkan tag ke sumber daya, atau memperbarui nilai tag yang ada.
- [ListTagsForResource](#)— Daftar tag untuk sumber daya.
- [UntagResource](#)— Menghapus tag dari sumber daya.

Tambahkan atau hapus tag dari sumber daya kapan saja. Untuk memperbarui nilai kunci tag yang ada, tambahkan tag baru dengan kunci yang sama dan nilai baru yang Anda inginkan ke sumber daya. Tindakan ini menggantikan nilai lama dengan yang baru. Meskipun dimungkinkan untuk menetapkan string kosong sebagai nilai tag, Anda tidak dapat menetapkan nilai null.

Menghapus sumber daya juga menghapus tag apa pun yang terkait dengannya.

Gunakan tag dengan kebijakan IAM

Gunakan tag sumber daya dalam kebijakan IAM Anda untuk mengontrol akses dan izin pengguna. Misalnya, kebijakan dapat memungkinkan pengguna untuk hanya membuat sumber daya yang memiliki tag tertentu yang dilampirkan. Kebijakan juga dapat membatasi pengguna untuk membuat atau memodifikasi sumber daya yang memiliki tanda tertentu.

Note

Jika Anda menggunakan tanda untuk mengizinkan atau menolak akses pengguna ke sumber daya, Anda harus menolak memberi pengguna kemampuan untuk menambahkan atau menghapus tanda tersebut untuk sumber daya yang sama. Jika tidak, pengguna dapat melewati batasan Anda dan mendapatkan akses ke sumber daya dengan memodifikasi tag-nya.

Anda dapat menggunakan kunci konteks kondisi berikut dan nilai dalam Condition elemen (juga disebut Condition blok) dari pernyataan kebijakan.

`aws:ResourceTag/tag-key: tag-value`

Izinkan atau tolak tindakan pada sumber daya dengan tanda tertentu.

`aws:RequestTag/tag-key: tag-value`

Mengharuskan tag tertentu digunakan (atau tidak digunakan) saat membuat atau memodifikasi sumber daya yang dapat diberi tag.

`aws:TagKeys: [tag-key, ...]`

Mengharuskan sekumpulan kunci tag tertentu digunakan (atau tidak digunakan) saat membuat atau memodifikasi sumber daya yang dapat diberi tag.

 Note

Kunci dan nilai konteks syarat dalam kebijakan IAM hanya berlaku pada tindakan yang memiliki sumber daya yang dapat ditandai sebagai parameter wajib. Misalnya, Anda dapat mengatur akses bersyarat berbasis tag untuk [ListAssets](#). Anda tidak dapat menyetel akses bersyarat berbasis tag [PutLoggingOptions](#) karena tidak ada sumber daya yang dapat diberi tag yang direferensikan dalam permintaan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke AWS sumber daya menggunakan tag sumber daya](#) dan [referensi kebijakan IAM JSON](#) di Panduan Pengguna IAM.

Contoh kebijakan IAM menggunakan tag

- [Lihat AWS IoT SiteWise aset berdasarkan tag](#)

Pemecahan masalah AWS IoT SiteWise

Gunakan informasi berikut untuk memecahkan masalah dengan AWS IoT SiteWise

Topik

- [Memecahkan masalah gateway Edge SiteWise](#)
- [Memecahkan masalah portal AWS IoT SiteWise](#)
- [Memecahkan masalah tindakan aturan AWS IoT SiteWise](#)
- [Memecahkan masalah operasi impor dan ekspor massal](#)

Memecahkan masalah gateway Edge SiteWise

Memecahkan masalah umum AWS IoT SiteWise Edge gateway dengan menjelajahi topik yang relevan.

Anda juga dapat melihat CloudWatch metrik yang dilaporkan oleh gateway SiteWise Edge untuk memecahkan masalah dengan konektivitas atau aliran data. Untuk informasi selengkapnya, lihat [Pantau AWS IoT SiteWise dengan CloudWatch metrik Amazon](#).

Topik

- [Konfigurasi dan akses log gateway SiteWise Edge](#)
- [Memecahkan masalah gateway SiteWise Edge](#)
- [Memecahkan masalah aplikasi AWS IoT SiteWise Edge pada Siemens Industrial Edge](#)
- [Memecahkan masalah integrasi sumber terbuka di Edge](#)
- [Memecahkan masalah AWS IoT Greengrass](#)

Konfigurasi dan akses log gateway SiteWise Edge

Sebelum Anda dapat melihat log gateway SiteWise Edge, Anda harus mengonfigurasi gateway SiteWise Edge Anda untuk mengirim log ke Amazon CloudWatch Logs atau menyimpan log di sistem file lokal.

- Gunakan CloudWatch Log jika Anda ingin menggunakan file log gateway SiteWise Edge AWS Management Console untuk melihat file log gateway Edge Anda. Untuk informasi selengkapnya, lihat [Gunakan CloudWatch Log Amazon](#).

- Gunakan log sistem file lokal jika Anda ingin menggunakan baris perintah atau perangkat lunak lokal untuk melihat file log gateway SiteWise Edge Anda. Untuk informasi selengkapnya, lihat [Gunakan log layanan di AWS IoT SiteWise](#).

Memecahkan masalah gateway SiteWise Edge

Gunakan informasi berikut untuk memecahkan masalah gateway SiteWise Edge.

Masalah

- [Tidak dapat menerapkan paket ke gateway SiteWise Edge](#)
- [AWS IoT SiteWise tidak menerima data dari server OPC UA](#)
- [Tidak ada data yang ditampilkan di dasbor](#)
- [“Tidak dapat menemukan atau memuat kelas utama” muncul di aws.iot. SiteWiseEdgePublisher log di/greengrass/v2/logs/kesalahan](#)
- [Saya melihat 'SESSION_TAKEN_OVER' atau 'com.aws.greengrass.mqttclient. MqttClient: Gagal mempublikasikan pesan melalui Spooler dan akan mencoba lagi. ' di log](#)
- [Saya melihat 'com.aws.greengrass.deployment. lotJobsHelper: Tidak ada pekerjaan penerapan yang ditemukan. ' atau 'Hasil penerapan sudah dilaporkan. ' di log](#)
- [Saya melihat status 'SYNC_FAILED' ketika mencoba mengonfigurasi pengaturan stempel waktu dalam grup properti pada sumber data OPC UA](#)
- [Tipe data yang dikonversi tidak termasuk](#)
- [Masalah toko kepercayaan](#)
- [Masalah instalasi berkemampuan proxy](#)

Tidak dapat menerapkan paket ke gateway SiteWise Edge

Jika AWS IoT Greengrass nucleus component (`aws.greengrass.Nucleus`) sudah kedaluwarsa, Anda mungkin tidak dapat menerapkan paket ke gateway SiteWise Edge Anda. Anda dapat menggunakan AWS IoT Greengrass V2 konsol untuk memutakhirkan komponen AWS IoT Greengrass nukleus.

Untuk memutakhirkan komponen AWS IoT Greengrass nukleus (konsol)

1. Navigasikan ke [konsol AWS IoT Greengrass](#) tersebut.

2. Di panel navigasi, di bawah AWS IoT Greengrass, pilih Deployment.
3. Dalam daftar Deployment, pilih deployment yang ingin Anda revisi.
4. Pilih Revise.
5. Pada halaman Tentukan target, pilih Selanjutnya.
6. Pada halaman Pilih komponen, di bawah Komponen publik, di kotak pencarian, masukkan **aws.greengrass.Nucleus**, lalu pilih AWS.GreenGrass.Nucleus.
7. Pilih Berikutnya.
8. Pada halaman Konfigurasi komponen, pilih Berikutnya.
9. Pada halaman Konfigurasi pengaturan lanjutan, pilih Berikutnya.
10. Di halaman Tinjau, pilih Deploy.

AWS IoT SiteWise tidak menerima data dari server OPC UA

Jika AWS IoT SiteWise aset Anda tidak menerima data yang dikirim oleh server OPC UA, Anda dapat mencari log gateway SiteWise Edge untuk memecahkan masalah. Cari `swPublisher` log tingkat info yang berisi pesan berikut.

```
Emitting diagnostic name=PublishError.SomeException
```

Berdasarkan jenis *SomeException* di log, gunakan jenis pengecualian berikut dan masalah terkait untuk memecahkan masalah gateway SiteWise Edge Anda:

- **ResourceNotFoundException**— Server OPC UA Anda mengirim data yang tidak cocok dengan alias properti untuk aset apa pun. Pengecualian ini dapat terjadi dalam dua kasus:
 - Alias properti Anda tidak sama persis dengan variabel OPC UA Anda, termasuk awalan sumber apa pun yang Anda tentukan. Periksa apakah alias properti dan awalan sumber Anda sudah benar.
 - Anda belum memetakan variabel OPC UA Anda ke properti aset. Untuk informasi selengkapnya, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).

Jika Anda sudah memetakan semua variabel OPC UA yang Anda inginkan AWS IoT SiteWise, Anda dapat memfilter variabel OPC UA mana yang dikirim gateway SiteWise Edge. Untuk informasi selengkapnya, lihat [Gunakan filter node OPC UA di SiteWise Edge](#).

- **InvalidRequestException**— Jenis data variabel OPC UA Anda tidak cocok dengan tipe data properti aset Anda. Misalnya, jika variabel OPC UA memiliki tipe data integer, properti aset terkait Anda

harus tipe data integer. Properti aset tipe ganda tidak dapat menerima nilai integer OPC UA. Untuk memperbaiki masalah ini, tentukan properti baru dengan tipe data yang benar.

- **TimestampOutOfRangeException**— Gateway SiteWise Edge Anda mengirim data yang berada di luar jangkauan yang AWS IoT SiteWise menerima. AWS IoT SiteWise menolak setiap titik data dengan stempel waktu lebih awal dari 7 hari di masa lalu atau lebih baru dari 5 menit di masa depan. Jika gateway SiteWise Edge Anda kehilangan daya atau koneksi ke AWS Cloud, Anda mungkin perlu menghapus cache gateway SiteWise Edge Anda.
- **ThrottlingException** atau **LimitExceededException**— Permintaan Anda melebihi kuota AWS IoT SiteWise layanan, seperti tingkat titik data yang dicerna atau tingkat permintaan untuk operasi API data properti aset. Periksa apakah konfigurasi Anda tidak melebihi [AWS IoT SiteWise kuota](#).

Tidak ada data yang ditampilkan di dasbor

Jika tidak ada data yang ditampilkan di dasbor Anda, konfigurasi Penerbit dan Sumber Data gateway SiteWise Edge mungkin tidak sinkron. Jika tidak sinkron, memperbarui nama sumber data dapat mempercepat sinkronisasi dari cloud ke tepi, memperbaiki kesalahan Keluar dari sinkronisasi.

Untuk memperbarui nama sumber data

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway SiteWise Edge yang terhubung ke dasbor.
4. Di bawah Sumber data, pilih Edit.
5. Pilih nama sumber baru, dan pilih Simpan untuk mengonfirmasi perubahan Anda.
6. Verifikasi perubahan Anda dengan mengonfirmasi nama sumber data telah diperbarui di tabel Sumber data.

“Tidak dapat menemukan atau memuat kelas utama” muncul di `aws.iot`.

SiteWiseEdgePublisher log di `/greengrass/v2/logs` kesalahan

Jika Anda melihat kesalahan ini, Anda mungkin perlu memperbarui versi java dari gateway SiteWise Edge Anda.

- Dari terminal, jalankan perintah berikut:

```
java -version
```

Versi java gateway SiteWise Edge Anda berjalan dengan akan muncul di bawah OpenJDK Runtime Environment. Anda akan melihat respons seperti berikut:

```
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)
```

Jika Anda menjalankan Java versi 11.0.20.8.1 Anda harus memperbarui paket IoT SiteWise Publisher ke versi 2.4.1 atau yang lebih baru. Hanya versi java 11.0.20.8.1 yang terpengaruh, lingkungan dengan versi java lainnya dapat terus menggunakan versi lama dari komponen IoT Publisher. SiteWise Untuk informasi selengkapnya tentang memperbarui paket komponen, lihat [Ubah versi paket komponen gateway SiteWise Edge](#).

Saya melihat 'SESSION_TAKEN_OVER' atau 'com.aws.greengrass.mqttclient.MqttClient: Gagal mempublikasikan pesan melalui Spooler dan akan mencoba lagi.' di log

Jika Anda melihat peringatan yang menyertakan SESSION_TAKEN_OVER atau kesalahan yang disertakan com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry. dalam log Anda di /greengrass/v2/logs/greengrass.log, Anda mungkin mencoba menggunakan file konfigurasi yang sama untuk beberapa gateway SiteWise Edge di beberapa perangkat. Setiap gateway SiteWise Edge memerlukan file konfigurasi unik untuk terhubung ke AWS akun Anda.

Saya melihat 'com.aws.greengrass.deployment.IotJobsHelper: Tidak ada pekerjaan penerapan yang ditemukan.' atau 'Hasil penerapan sudah dilaporkan.' di log

Jika Anda melihat com.aws.greengrass.deployment.IotJobsHelper: No deployment job found. atau Deployment result already reported. dalam log Anda di /greengrass/v2/logs/greengrass.log, Anda mungkin mencoba untuk menggunakan kembali file konfigurasi yang sama.

Ada beberapa solusi:

- Jika Anda ingin menggunakan kembali file konfigurasi, lakukan hal berikut:

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.

2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway SiteWise Edge yang ingin Anda gunakan kembali.
4. Pilih tab Pembaruan.
5. Pilih versi Publisher yang berbeda dan pilih Deploy.

Ikuti langkah-langkah [Buat gateway untuk Siemens Industrial Edge](#) untuk membuat file konfigurasi baru.

Saya melihat status 'SYNC_FAILED' ketika mencoba mengonfigurasi pengaturan stempel waktu dalam grup properti pada sumber data OPC UA

Saat AWS IoT SiteWise memperbarui komponen kolektor OPC UA untuk AWS IoT Greengrass versi 2.5.0, kami memperkenalkan opsi konfigurasi stempel waktu baru. Anda dapat menggunakan stempel waktu dari perangkat Anda, atau stempel waktu dari server. Versi lama dari komponen kolektor OPC UA tidak mendukung opsi ini dan gagal disinkronkan.

Ada dua cara untuk mengatasi status sinkronisasi sumber data yang gagal. Cara yang disarankan adalah dengan memutakhirkan komponen kolektor IoT SiteWise OPC UA ke versi 2.5.0 atau lebih tinggi. Atau, Anda dapat terus menggunakan versi komponen kolektor OPC UA yang lebih lama, jika Anda menyetel stempel waktu ke. `Source` Untuk mempelajari cara memutakhirkan komponen kolektor IoT SiteWise OPC UA, lihat. [Perbarui versi AWS IoT SiteWise komponen](#) Kami merekomendasikan menggunakan versi terbaru dari semua komponen.

 Note

Tidak ada gangguan data saat status sinkronisasi sumber data gagal. Sumber data terus mengalir ke AWS IoT SiteWise. Konfigurasi tidak disinkronkan dengan komponen kolektor SiteWise IoT OPC UA pada penerapan Anda. AWS IoT Greengrass V2

Untuk mengubah konfigurasi stempel waktu untuk grup properti

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih gateway untuk diedit.
4. Di bagian Sumber data, pilih sumber data dengan status sinkronisasi gagal, lalu pilih Edit.
5. Perluas Konfigurasi lanjutan, lalu perluas Pengaturan grup.

6. Di Timestamp, pilih Sumber. Memilih Sumber menghapus `timestampToReturn` properti dari konfigurasi. Pengaturan ini memungkinkan pengumpulan stempel waktu sumber data dari perangkat Anda secara default, memungkinkan sumber data untuk disinkronkan dengan komponen kolektor IoT SiteWise OPC UA.
7. Pilih Simpan.

Tipe data yang dikonversi tidak termasuk

Jika Anda melihat kesalahan saat mengonversi tipe data OPC UA yang tidak didukung menjadi string AWS IoT SiteWise, ada beberapa kemungkinan alasan:

- Tipe data yang Anda coba konversi adalah tipe data yang kompleks. Tipe data yang kompleks tidak didukung.
- Saat menggunakan Destinasi sebagai AWS IoT SiteWise Buffered menggunakan Amazon S3, nilai string penuh dipertahankan dalam file yang didorong ke bucket Amazon S3. Saat Anda kemudian memasukkan data ke dalam AWS IoT SiteWise, nilai string penuh yang lebih panjang dari 1024 byte ditolak.

Masalah toko kepercayaan

Jika Anda mengalami masalah yang terkait dengan toko kepercayaan di SiteWise Edge, pertimbangkan langkah-langkah pemecahan masalah berikut:

- Verifikasi bahwa sertifikat CA AWS IoT Greengrass root ada dan diformat dengan benar di toko kepercayaan yang sesuai
- Pastikan KeyStore kata sandi Java diatur dengan benar dan dapat diakses oleh komponen SiteWise Edge
- Periksa apakah sertifikat khusus apa pun (seperti untuk proxy HTTPS) dalam format yang benar (biasanya PEM) dan diimpor dengan benar ke toko kepercayaan
- Konfirmasikan bahwa toko kepercayaan memiliki izin file yang benar dan dapat diakses oleh proses SiteWise Edge
- Tinjau log SiteWise Edge untuk setiap kesalahan SSL/TLS terkait, yang mungkin mengindikasikan masalah penyimpanan kepercayaan
- Uji SSL/TLS koneksi secara independen menggunakan alat seperti `openssl` memverifikasi fungsionalitas toko kepercayaan

Masalah instalasi berkemampuan proxy

Jika Anda mengalami masalah selama proses konfigurasi proxy, pertimbangkan langkah-langkah pemecahan masalah berikut:

- Verifikasi bahwa URL proxy diformat dengan benar dan menyertakan skema yang tepat (<http://>atau<https://>)
- Pastikan bahwa setiap kredensi proxy dikodekan URL jika mengandung karakter khusus
- Konfirmasikan bahwa daftar no-proxy mencakup semua alamat lokal dan titik akhir AWS layanan yang diperlukan
- Untuk proxy HTTPS, verifikasi bahwa sertifikat CA yang disediakan dalam format PEM
- Tinjau log instalasi untuk pesan kesalahan tertentu yang mungkin menunjukkan sumber masalah
- Uji koneksi proxy secara independen untuk memastikannya berfungsi dengan benar

Memecahkan masalah aplikasi AWS IoT SiteWise Edge pada Siemens Industrial Edge

Untuk memecahkan masalah aplikasi AWS IoT SiteWise Edge di Siemens Industrial Edge perangkat Anda, Anda dapat mengakses log untuk aplikasi melalui portal Siemens Industrial Edge Management atau Siemens Industrial Edge Device (IED). Untuk informasi selengkapnya, lihat [Mengunduh Log](#) di dokumentasi Siemens.

Data saya tidak ditampilkan di AWS IoT SiteWise

- Pastikan tidak ada masalah dengan Databus pengguna Anda dan ikon tanda centang untuk Databus_Configuration berwarna hijau daripada abu-abu.
- Anda mungkin tidak Siemens Industrial Edge Management menjalankan versi yang berisi Secure Storage. Tingkatkan versi Siemens OS Anda. Untuk informasi selengkapnya, lihat [Siemens Secure Storage dan aplikasi AWS IoT SiteWise Edge](#).

Saya melihat 'File konfigurasi hilang AWS_REGION 'di log

Jika Anda melihat Config file missing AWS_REGION di log Siemens, JSON dari file konfigurasi telah rusak. Anda harus membuat file konfigurasi baru. Ikuti langkah-langkah [Buat gateway untuk Siemens Industrial Edge](#) untuk membuat file konfigurasi baru.

Saya melihat pesan kesalahan 'Out of sync' pada konfigurasi gateway Edge

Jika Anda melihat pesan Out of sync kesalahan di Siemens Industrial Edge gateway setelah penerapan selesai, itu berarti komponen penerbit SiteWise IoT tidak sinkron dengan gateway Anda. Komponen SiteWise penerbit IoT bekerja di latar belakang Siemens Industrial Edge gateway untuk menyediakan fungsionalitas topik MQTT. Kami meningkatkan Siemens Industrial Edge gateway untuk menggunakan `iotsitewise:publisher:3` namespace kemampuan daripada `iotsitewise:publisher:2`. Anda dapat memperbarui ke versi terbaru penerbit untuk mengatasi masalah ini.

Untuk meningkatkan ke versi terbaru dari penerbit IoT SiteWise

1. Navigasikan ke [konsol AWS IoT SiteWise](#) tersebut.
2. Di panel navigasi, pilih gateway Edge.
3. Pilih Siemens Industrial Edge gateway untuk diedit.
4. Di bagian kemampuan Edge, pilih versi perangkat lunak Lihat.
5. Pilih versi terbaru penerbit IoT di bawah menu SiteWise tarik-turun Penerbit.
6. Pilih Selesai.

Memecahkan masalah integrasi sumber terbuka di Edge

Bagian ini memberikan solusi untuk masalah umum yang mungkin Anda temui saat mengintegrasikan alat sumber terbuka dengan SiteWise Edge.

Note

Node-red®, InfluxDB®, dan Grafana® bukan vendor atau pemasok untuk Edge. SiteWise

Masalah koneksi

Node-red tidak dapat terhubung ke broker MQTT

Verifikasi bahwa broker MQTT berjalan dan dapat diakses pada port yang ditentukan. Periksa konfigurasi jaringan Anda dan pastikan bahwa alamat broker sudah benar.

Untuk memverifikasi status broker MQTT, jalankan:

```
docker ps | grep emqx
```

Kesalahan koneksi InfluxDB

Pastikan token otentikasi Anda valid dan Anda telah menentukan nama organisasi dan bucket yang benar. Periksa apakah InfluxDB berjalan dan dapat diakses.

Untuk memverifikasi status InfluxDB, jalankan:

```
curl -I http://localhost:8086
```

Grafana tidak dapat terhubung ke InfluxDB

Verifikasi bahwa konfigurasi sumber data InfluxDB di Grafana sudah benar, termasuk URL, token otentikasi, organisasi, dan bucket.

Masalah aliran data

Tidak ada data yang muncul di AWS IoT SiteWise

Periksa apakah alias properti Anda dalam alur Node-merah cocok dengan format yang diharapkan. Verifikasi bahwa struktur topik MQTT sudah benar dan gateway SiteWise Edge dikonfigurasi dengan benar untuk menerima data dari broker MQTT.

Tidak ada data SiteWise Edge yang disimpan di InfluxDB

Verifikasi bahwa alur retensi Node-Red dikonfigurasi dengan benar dan node penulis InfluxDB memiliki pengaturan bucket dan pengukuran yang tepat. Periksa output debug Node-Red untuk kesalahan apa pun.

Kesalahan format data

Pastikan transformasi data Anda berfungsi dengan benar mengonversi data antar format. Gunakan node debug Node-Red untuk memeriksa data pada setiap tahap aliran.

Masalah kinerja

Penggunaan CPU atau memori yang tinggi

Pantau penggunaan sumber daya dan sesuaikan konfigurasi komponen Anda sesuai kebutuhan. Pertimbangkan untuk mengurangi frekuensi pengumpulan data atau menerapkan penyaringan data untuk mengurangi beban pemrosesan.

Untuk memantau penggunaan sumber daya, jalankan:

```
docker stats
```

Pemuatan dasbor Grafana lambat

Optimalkan kueri InfluxDB Anda dan pertimbangkan untuk menambahkan batasan rentang waktu ke panel dasbor Anda. Kurangi jumlah titik data yang ditampilkan dengan menggunakan fungsi agregasi yang sesuai.

Pencatatan dan diagnostik

Untuk memecahkan masalah, periksa log untuk setiap komponen:

Log merah simpul

Lihat log di konsol Node-merah atau jalankan:

```
docker logs node-red
```

Log InfluxDB

Akses log dengan menjalankan:

```
docker logs influxdb
```

Log Grafana

Lihat log dengan menjalankan:

```
docker logs grafana
```

SiteWise Log tepi

Periksa log gateway SiteWise Edge untuk masalah koneksi dan pemrosesan data MQTT. Untuk informasi selengkapnya, lihat [Memecahkan masalah gateway Edge SiteWise](#).

Memecahkan masalah AWS IoT Greengrass

Untuk menemukan solusi untuk banyak masalah saat mengonfigurasi atau menerapkan gateway SiteWise Edge AWS IoT Greengrass, lihat [Pemecahan Masalah AWS IoT Greengrass](#) di Panduan Pengembang AWS IoT Greengrass.

Memecahkan masalah portal AWS IoT SiteWise

Memecahkan masalah umum dengan portal Anda AWS IoT SiteWise.

Pengguna dan administrator tidak dapat mengakses portal AWS IoT SiteWise

Jika pengguna atau administrator tidak dapat mengakses AWS IoT SiteWise portal Anda, Anda mungkin telah membatasi izin dalam kebijakan terlampir AWS Identity and Access Management (IAM) yang mencegah login berhasil.

Lihat contoh kebijakan IAM berikut yang akan mengakibatkan kegagalan login:

Note

Setiap kebijakan IAM terlampir yang menyertakan "Condition" elemen akan menyebabkan kegagalan login.

Contoh 1: Kondisi di sini adalah IP terbatas, dan ini akan menyebabkan kegagalan login.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*",
    "Condition": {
        "IpAddress": {
            "aws:SourceIp": [
                "REPLACESAMPLEIP"
            ]
        }
    }
}
]
}

```

Contoh 2: Kondisi di sini adalah tag yang disertakan, dan ini akan menyebabkan kegagalan login.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/project": "*"
        }
      }
    }
  ]
}

```

Saat menambahkan pengguna atau administrator ke portal, hindari membuat kebijakan IAM yang membatasi izin pengguna, seperti IP terbatas. Kebijakan terlampir dengan izin terbatas tidak akan dapat terhubung ke AWS IoT SiteWise portal.

Memecahkan masalah tindakan aturan AWS IoT SiteWise

Untuk memecahkan masalah tindakan AWS IoT SiteWise aturan Anda AWS IoT Core, Anda dapat melakukan salah satu prosedur berikut:

- Konfigurasi CloudWatch Log Amazon
- Konfigurasi tindakan kesalahan publikasi ulang untuk aturan Anda

Kemudian, bandingkan pesan kesalahan dengan kesalahan dalam topik ini untuk memecahkan masalah Anda.

Topik

- [Konfigurasi AWS IoT Core log](#)
- [Konfigurasi tindakan kesalahan penerbitan ulang](#)
- [Memecahkan masalah aturan](#)
- [Memecahkan masalah aturan \(AWS IoT SiteWise\)](#)
- [Memecahkan masalah aturan \(DynamoDB\)](#)

Konfigurasi AWS IoT Core log

Anda dapat mengonfigurasi AWS IoT untuk mencatat berbagai tingkat informasi ke CloudWatch Log.

Untuk mengonfigurasi dan mengakses CloudWatch Log

1. Untuk mengonfigurasi pencatatan AWS IoT Core, lihat [Pemantauan dengan CloudWatch Log](#) di Panduan AWS IoT Pengembang.
2. Navigasikan ke [konsol CloudWatch](#) tersebut.
3. Pada panel navigasi, pilih Grup log.
4. Pilih grup AWSIoTLog.
5. Pilih aliran log terbaru. Secara default, CloudWatch menampilkan aliran log terbaru terlebih dahulu.
6. Pilih entri log untuk memperluas pesan log. Entri log Anda mungkin terlihat seperti tangkapan layar berikut.

CloudWatch > Log Groups > AWSIoTLogs > 9ca6614a-00fc-4f9e-8100-5c2a34918e90_123456789012_0

Expand all ☒ Row ☐ Text   

Filter events all 2020-02-10 (19:36:11) ▾

Time (UTC +00:00)	Message
2020-02-11	No older events found at the moment. Retry .
19:36:11	2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWise
	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. Retry .

7. Bandingkan pesan kesalahan dengan kesalahan dalam topik ini untuk memecahkan masalah Anda.

Konfigurasi tindakan kesalahan penerbitan ulang

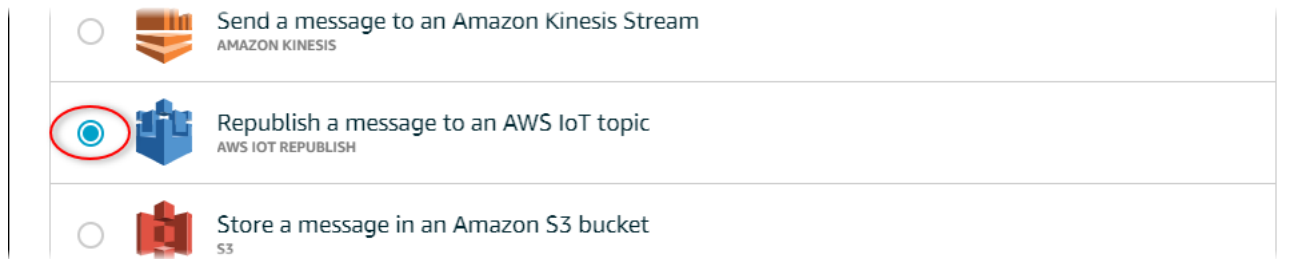
Anda dapat mengonfigurasi tindakan kesalahan pada aturan Anda untuk menangani pesan kesalahan. Dalam prosedur ini, Anda mengonfigurasi tindakan aturan penerbitan ulang sebagai tindakan kesalahan untuk melihat pesan kesalahan di klien pengujian MQTT.

Note

Tindakan kesalahan penerbitan ulang hanya menghasilkan log ERROR level yang setara. Jika Anda ingin lebih banyak log verbose, Anda harus [mengkonfigurasi CloudWatch Log](#).

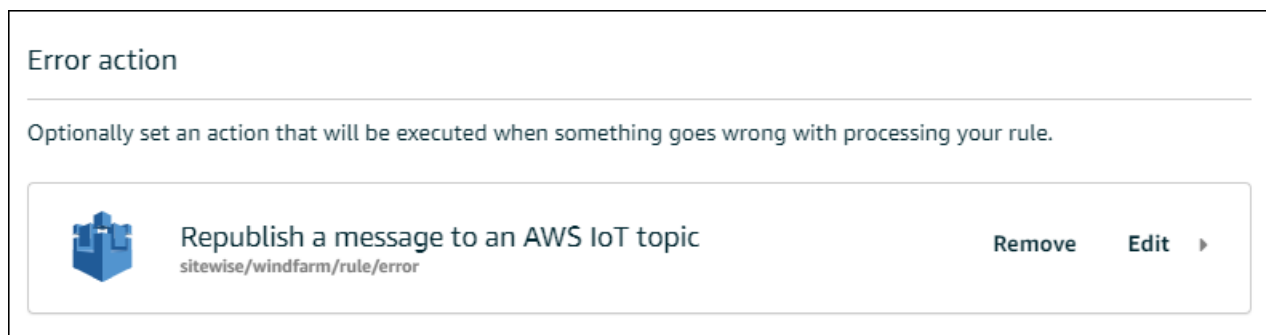
Untuk menambahkan tindakan kesalahan penerbitan ulang ke aturan

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi kiri, pilih Bertindak lalu pilih Aturan.
3. Pilih aturan Anda.
4. Di bawah Tindakan kesalahan, pilih Tambah tindakan.
5. Pilih Menerbitkan ulang pesan ke suatu AWS IoT topik.



6. Pilih Konfigurasi tindakan di bagian bawah halaman.
7. Di Topik, masukkan topik unik (misalnya, **sitewise/windfarm/rule/error**). AWS IoT Core akan menerbitkan ulang pesan kesalahan ke topik ini.
8. Pilih Pilih untuk memberikan AWS IoT Core akses untuk melakukan tindakan kesalahan.
9. Pilih Pilih di samping peran yang Anda buat untuk aturan.
10. Pilih Perbarui Peran untuk menambahkan izin tambahan ke peran.
11. Pilih Tambahkan tindakan.

Tindakan kesalahan aturan Anda akan terlihat mirip dengan tangkapan layar berikut.



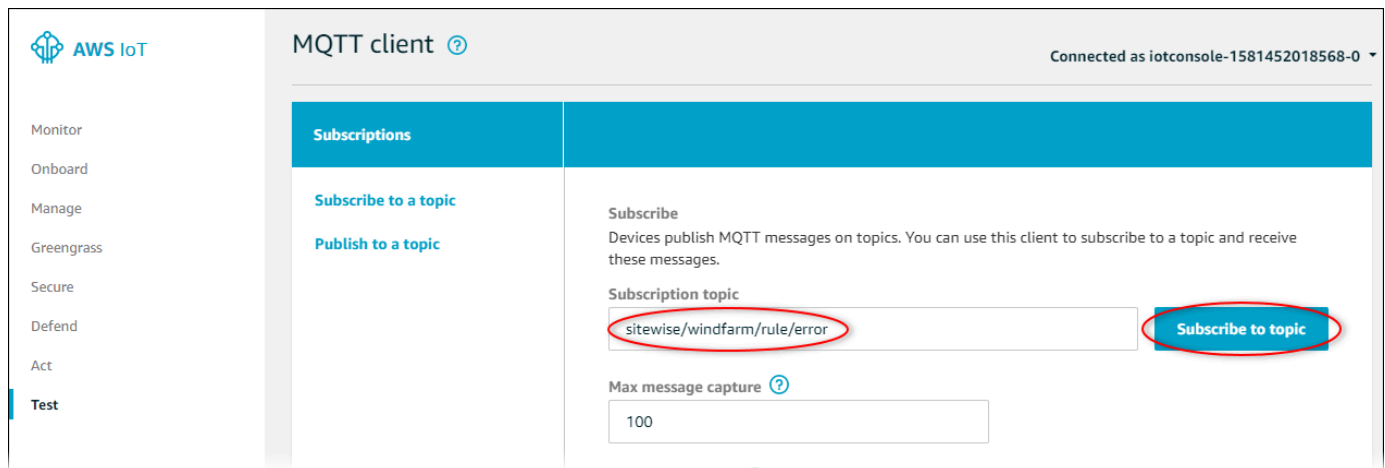
12. Pilih panah belakang di kiri atas konsol untuk kembali ke rumah AWS IoT konsol.

Setelah mengatur tindakan kesalahan penerbitan ulang, Anda dapat melihat pesan kesalahan di klien pengujian MQTT di. AWS IoT Core

Dalam prosedur berikut, Anda berlangganan topik kesalahan di klien pengujian MQTT. Di klien pengujian MQTT, Anda dapat menerima pesan kesalahan aturan untuk memecahkan masalah.

Untuk berlangganan topik tindakan kesalahan

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di halaman navigasi kiri, pilih Uji untuk membuka klien pengujian MQTT.
3. Di bidang Subscription topic, masukkan topik error yang Anda konfigurasi sebelumnya (misalnya, **sitewise/windfarm/rule/error**) dan pilih Subscribe to topic.



- Perhatikan pesan kesalahan muncul dan kemudian perluas `failures` array dalam pesan kesalahan apa pun.

Selanjutnya, bandingkan pesan kesalahan dengan kesalahan dalam topik ini untuk memecahkan masalah Anda.

Memecahkan masalah aturan

Gunakan informasi berikut untuk memecahkan masalah aturan.

Masalah

- [Kesalahan: Anggota harus berada dalam 604800 detik sebelum dan 300 detik setelah stempel waktu saat ini](#)
- [Kesalahan: Nilai properti tidak cocok dengan tipe data <type>](#)
- [Kesalahan: Pengguna: <role-arn> tidak berwenang untuk melakukan: iotsitewise: pada sumber daya BatchPutAssetPropertyValue](#)
- [Kesalahan: iot.amazonaws.com tidak dapat melakukan: sts: on resource: AssumeRole <role-arn>](#)
- [Info: Tidak ada permintaan yang dikirim. PutAssetPropertyValueEntries kosong setelah melakukan templat substitusi.](#)

Kesalahan: Anggota harus berada dalam 604800 detik sebelum dan 300 detik setelah stempel waktu saat ini

Stempel waktu Anda lebih tua dari 7 hari atau lebih baru dari 5 menit, dibandingkan dengan waktu epoch Unix saat ini. Coba yang berikut ini:

- Periksa apakah stempel waktu Anda dalam waktu Unix epoch (UTC). Jika Anda memberikan stempel waktu dengan zona waktu yang berbeda, Anda menerima kesalahan ini.
- Periksa apakah stempel waktu Anda dalam hitungan detik. AWS IoT SiteWise mengharapkan stempel waktu dibagi menjadi waktu dalam detik (dalam waktu epoch Unix) dan diimbangi dalam nanodetik.
- Periksa apakah Anda mengunggah data yang diberi stempel waktu selambat-lambatnya 7 hari sebelumnya.

Kesalahan: Nilai properti tidak cocok dengan tipe data <type>

Entri dalam tindakan aturan Anda memiliki tipe data yang berbeda dari properti aset target. Misalnya, properti aset target Anda adalah `DOUBLE` dan tipe data yang Anda pilih adalah Integer atau Anda meneruskan nilainya. `integerValue` Coba yang berikut ini:

- Jika Anda mengonfigurasi aturan dari AWS IoT konsol, periksa apakah Anda memilih tipe Data yang benar untuk setiap entri.
- Jika Anda mengonfigurasi aturan dari API atau AWS Command Line Interface (AWS CLI), periksa apakah `value` objek Anda menggunakan bidang tipe yang benar (misalnya, `doubleValue` untuk `DOUBLE` properti).

Kesalahan: Pengguna: <role-arn>tidak berwenang untuk melakukan: `iotsitewise:BatchPutAssetPropertyValue` pada sumber daya `BatchPutAssetPropertyValue`

Aturan Anda tidak diizinkan untuk mengakses properti aset target, atau properti aset target tidak ada. Coba yang berikut ini:

- Periksa apakah alias properti Anda benar dan bahwa Anda memiliki properti aset dengan alias properti yang diberikan. Untuk informasi selengkapnya, lihat [Mengelola aliran data untuk AWS IoT SiteWise](#).
- Pastikan aturan Anda memiliki peran dan peran tersebut mengizinkan `iotsitewise:BatchPutAssetPropertyValue` izin ke properti aset yang ditargetkan, seperti melalui hierarki aset target. Untuk informasi selengkapnya, lihat [Berikan AWS IoT akses yang diperlukan](#).

Kesalahan: `iot.amazonaws.com` tidak dapat melakukan: `sts: on resource: AssumeRole <role-arn>`

Pengguna Anda tidak diizinkan untuk mengambil peran pada aturan Anda di AWS Identity and Access Management (IAM).

Periksa apakah pengguna Anda diizinkan `iam:PassRole` izin untuk peran pada aturan Anda. Untuk informasi selengkapnya, lihat [Meneruskan izin peran](#) di Panduan AWS IoT Pengembang.

Info: Tidak ada permintaan yang dikirim. `PutAssetPropertyValueEntries` kosong setelah melakukan templat substitusi.

 Note

Pesan ini adalah log INFO level.

Permintaan Anda harus memiliki setidaknya satu entri dengan semua parameter yang diperlukan.

Periksa apakah parameter aturan Anda, termasuk templat substitusi, menghasilkan nilai yang tidak kosong. Templat substitusi tidak dapat mengakses nilai yang ditentukan dalam AS klausa dalam pernyataan kueri aturan Anda. Untuk informasi selengkapnya, lihat [Templat substitusi](#) di Panduan AWS IoT Pengembang.

Memecahkan masalah aturan ()AWS IoT SiteWise

Ikuti langkah-langkah dalam prosedur ini untuk memecahkan masalah aturan Anda jika data penggunaan CPU dan memori tidak muncul AWS IoT SiteWise seperti yang diharapkan. Dalam prosedur ini, Anda mengonfigurasi tindakan aturan penerbitan ulang sebagai tindakan kesalahan untuk melihat pesan kesalahan di klien pengujian MQTT. Anda juga dapat mengonfigurasi logging ke CloudWatch Log untuk memecahkan masalah. Untuk informasi selengkapnya, lihat [Memecahkan masalah tindakan aturan AWS IoT SiteWise](#).

Untuk menambahkan tindakan kesalahan penerbitan ulang ke aturan

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi kiri, pilih Perutean pesan, lalu pilih Aturan.
3. Pilih aturan yang Anda buat sebelumnya dan pilih Edit.

4. Di bawah Tindakan kesalahan - opsional, pilih Tambahkan tindakan kesalahan.
5. Pilih Menerbitkan ulang pesan ke suatu AWS IoT topik.
6. Di Topik, masukkan jalur ke kesalahan Anda (misalnya, **sitewise/rule/tutorial/error**). AWS IoT Core akan menerbitkan ulang pesan kesalahan ke topik ini.
7. Pilih peran yang Anda buat sebelumnya (misalnya, SiteWiseTutorialDeviceRuleRole).
8. Pilih Perbarui.

Setelah mengatur tindakan kesalahan penerbitan ulang, Anda dapat melihat pesan kesalahan di klien pengujian MQTT di AWS IoT Core

Dalam prosedur berikut, Anda berlangganan topik kesalahan di klien pengujian MQTT.

Untuk berlangganan topik tindakan kesalahan

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di halaman navigasi kiri, pilih klien pengujian MQTT untuk membuka klien pengujian MQTT.
3. Di bidang Filter topik, masukkan **sitewise/rule/tutorial/error** dan pilih Berlangganan.

Saat pesan kesalahan muncul, lihat `failures` larik dalam pesan kesalahan apa pun untuk mendiagnosis masalah. Untuk informasi selengkapnya tentang kemungkinan masalah dan cara mengatasinya, lihat [Memecahkan masalah tindakan aturan AWS IoT SiteWise](#).

Jika kesalahan tidak muncul, periksa apakah aturan Anda diaktifkan dan Anda berlangganan topik yang sama dengan yang Anda konfigurasi dalam tindakan kesalahan penerbitan ulang. Jika kesalahan masih tidak muncul setelah Anda melakukannya, periksa apakah skrip perangkat berjalan dan memperbarui bayangan perangkat dengan sukses.

Note

Anda juga dapat berlangganan topik pembaruan bayangan perangkat Anda untuk melihat muatan yang diuraikan AWS IoT SiteWise tindakan Anda. Untuk melakukannya, berlangganan topik berikut.

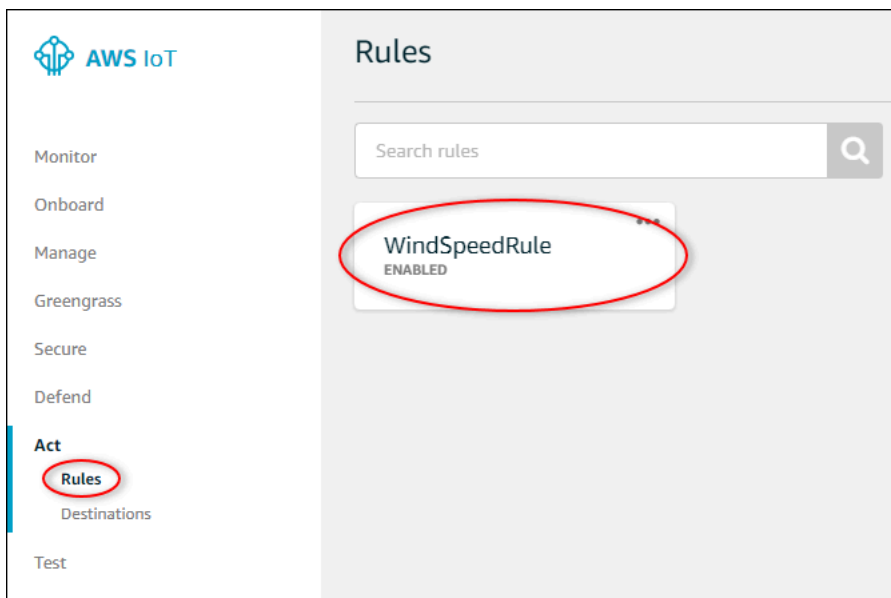
```
$aws/things/+/shadow/update/accepted
```

Memecahkan masalah aturan (DynamoDB)

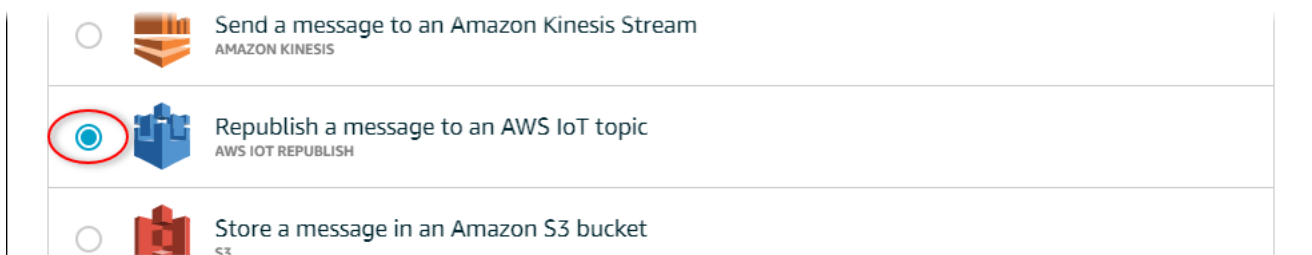
Ikuti langkah-langkah dalam prosedur ini untuk memecahkan masalah aturan Anda jika data aset demo tidak muncul di tabel DynamoDB seperti yang diharapkan. Dalam prosedur ini, Anda mengonfigurasi tindakan aturan penerbitan ulang sebagai tindakan kesalahan untuk melihat pesan kesalahan di klien pengujian MQTT. Anda juga dapat mengonfigurasi logging ke CloudWatch Log untuk memecahkan masalah. Untuk informasi selengkapnya, lihat [Memantau dengan CloudWatch Log](#) di Panduan AWS IoT Pengembang.

Untuk menambahkan tindakan kesalahan penerbitan ulang ke aturan

1. Navigasikan ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi kiri, pilih Bertindak lalu pilih Aturan.
3. Pilih aturan yang Anda buat sebelumnya.



4. Di bawah Tindakan kesalahan, pilih Tambah tindakan.
5. Pilih Menerbitkan ulang pesan ke suatu AWS IoT topik.



6. Pilih Konfigurasi tindakan di bagian bawah halaman.

7. Di Topik, masukkan **windspeed/error**. AWS IoT Core akan menerbitkan ulang pesan kesalahan ke topik ini.

The screenshot shows the 'Configure action' dialog for the 'Republish a message to an AWS IoT topic' action. The dialog has a blue header with the title 'Configure action'. Below the header, there is a section with the AWS IoT Republish icon and the title 'Republish a message to an AWS IoT topic'. A description states: 'This action will republish the message to another AWS IoT topic.' The configuration section includes a '*Topic' field with a help icon, where 'windspeed/error' is entered and circled in red. Below this is the 'Quality of Service' section with a help icon, showing two radio button options: '0 - The message is delivered zero or more times.' (selected) and '1 - The message is delivered one or more times.' Below the configuration section, there is a section titled 'Choose or create a role to grant AWS IoT access to perform this action.' It contains a 'No role selected' button, a 'Create Role' button, and a 'Select' button, which is circled in red. At the bottom of the dialog, there are 'Cancel' and 'Add action' buttons.

8. Pilih Pilih untuk memberikan akses AWS IoT Inti untuk melakukan tindakan kesalahan menggunakan peran yang Anda buat sebelumnya.
9. Pilih Pilih di sebelah peran Anda.

The screenshot shows the 'Choose or create a role to grant AWS IoT access to perform this action.' dialog. It features a search bar with the placeholder text 'Search for IAM roles'. Below the search bar, a list of roles is displayed, with 'WindSpeedDataRole' selected and circled in red. To the right of the role list, there are three buttons: 'Refresh', 'Create Role', and 'Close'. The 'Select' button from the previous screenshot is also visible, circled in red.

10. Pilih Perbarui Peran untuk menambahkan izin tambahan ke peran.

1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

WindSpeedDataRole	Update Role	Create Role	Select

Cancel

Add action

11. Pilih Tambahkan tindakan untuk menyelesaikan penambahan tindakan kesalahan.
12. Pilih panah belakang di kiri atas konsol untuk kembali ke rumah konsol AWS IoT Core.

Setelah mengatur tindakan kesalahan penerbitan ulang, Anda dapat melihat pesan kesalahan di klien pengujian MQTT di Core. AWS IoT

Dalam prosedur berikut, Anda berlangganan topik kesalahan di klien pengujian MQTT.

Untuk berlangganan topik tindakan kesalahan

1. Di halaman navigasi kiri konsol AWS IoT inti, pilih Uji.
2. Di bidang Subscription topic, masukkan **windspeed/error** dan pilih Subscribe to topic.

AWS IoT

MQTT client

Connected as iotconsole-1578083417073-0

Subscriptions

Subscribe to a topic

Publish to a topic

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

windspeed/error

Subscribe to topic

Max message capture

100

3. Perhatikan pesan kesalahan muncul dan jelajahi failures larik dalam pesan kesalahan untuk mendiagnosis masalah umum berikut:
- Kesalahan ketik dalam pernyataan kueri aturan
 - Izin peran tidak memadai

Jika kesalahan tidak muncul, periksa apakah aturan Anda diaktifkan dan Anda berlangganan topik yang sama dengan yang Anda konfigurasi dalam tindakan kesalahan penerbitan ulang. Jika kesalahan masih tidak muncul, periksa apakah aset ladang angin demo Anda masih ada dan Anda mengaktifkan notifikasi pada properti kecepatan angin. Jika aset demo Anda kedaluwarsa dan hilang AWS IoT SiteWise, Anda dapat membuat demo baru dan memperbarui pernyataan kueri aturan untuk mencerminkan model aset dan properti IDs yang diperbarui.

Memecahkan masalah operasi impor dan ekspor massal

Untuk menangani dan mendiagnosis kesalahan yang dihasilkan selama tugas transfer, lihat AWS IoT TwinMaker `GetMetadataTransferJobAPI`:

1. Setelah membuat dan menjalankan tugas transfer, panggil `GetMetadataTransferJobAPI`:

```
aws iottwinmaker get-metadata-transfer-job \  
--metadata-transfer-job-id your_metadata_transfer_job_id \  
--region us-east-1
```

2. Keadaan pekerjaan berubah menjadi salah satu negara di bawah ini:
 - DISELESAIKAN
 - DIBATALKAN
 - ERROR
3. `GetMetadataTransferJobAPI` mengembalikan [MetadataTransferJobProgress](#) objek.
4. `MetadataTransferJobProgress` objek berisi parameter berikut:
 - `FailedCount`: Menunjukkan jumlah aset yang gagal selama proses transfer.
 - `SkippedCount`: Menunjukkan jumlah aset yang dilewati selama proses transfer.
 - `SucceededCount`: Menunjukkan jumlah aset yang berhasil selama proses transfer.
 - `TotalCount`: Menunjukkan jumlah total aset yang terlibat dalam proses transfer.
5. Selain itu, elemen `reportURL` dikembalikan oleh panggilan API, yang berisi URL yang telah ditandatangani sebelumnya. Jika pekerjaan transfer Anda memiliki kesalahan yang perlu diselidiki, Anda dapat mengunduh laporan kesalahan lengkap di URL ini.

AWS IoT SiteWise titik akhir dan kuota

Bagian berikut menjelaskan titik akhir dan kuota untuk AWS IoT SiteWise

Topik

- [AWS IoT SiteWise titik akhir](#)
- [AWS IoT SiteWise kuota](#)

AWS IoT SiteWise titik akhir

Referensi Umum AWS Panduan ini mencantumkan AWS IoT SiteWise titik akhir untuk sebuah Akun AWS. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise titik akhir dan kuota](#) di Panduan Referensi Umum AWS

AWS IoT SiteWise kuota

Tabel berikut menjelaskan kuota di AWS IoT SiteWise. Untuk informasi selengkapnya tentang kuota dan cara meminta kenaikan kuota, lihat [kuota AWS layanan](#) di Referensi Umum AWS Untuk informasi lebih lanjut tentang AWS IoT SiteWise kuota, lihat [kuota AWS IoT SiteWise layanan](#) di Referensi Umum AWS

Kuota untuk AWS IoT SiteWise aset dan model aset

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah model aset di masing-masing Wilayah AWS untuk masing-masing Akun AWS	Jumlah maksimum model aset yang dapat Anda buat di Wilayah AWS for a Akun AWS.	1000	Ya
Jumlah aset dalam setiap model aset	Jumlah maksimum aset yang dapat Anda buat untuk setiap model aset.	10.000	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah aset anak di setiap aset induk	Jumlah maksimum aset anak yang dapat Anda kaitkan dengan aset induk.	2000	Ya
Kedalaman pohon hierarki model aset	Kedalaman pohon hierarki aset maksimum untuk model aset.	30	Ya
Jumlah definisi hierarki di setiap model aset	Jumlah maksimum definisi hierarki yang dapat Anda miliki dalam model aset.	30	Ya
Jumlah properti di tingkat root di setiap model aset	Jumlah maksimum <code>assetMode</code> <code>lProperties</code> untuk setiap model aset. Hitungan ini tidak termasuk <code>composite</code> <code>ModelProperties</code> . Kuota ini juga berlaku untuk setiap aset unik yang dibuat dari model aset ini.	500	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah properti dalam model aset	Jumlah maksimum properti dari model aset jenis ASSET_MODEL atau COMPONENT_MODEL . Angka ini ditentukan dengan menggabungkan properti model aset akar dan model komposit yang disertakan component-model-based atau sebaris. Kuota ini juga berlaku untuk setiap aset unik yang dibuat dari model aset ini.	5000	Ya
Jumlah properti di setiap model komposit	Jumlah maksimum properti yang diizinkan untuk model komposit. Juga, jumlah maksimum properti yang diizinkan untuk model aset tipe COMPONENT_MODEL .	100	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Kedalaman pohon properti dalam model aset	Misalnya, model dengan properti transformasi C yang mengkonsumsi properti transformasi B yang mengkonsumsi properti pengukuran A memiliki kedalaman 3.	10	Tidak
Jumlah model aset di setiap pohon hierarki	Jumlah maksimum model aset yang dapat Anda sertakan dalam satu pohon hierarki.	100	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah properti yang bergantung langsung untuk model aset	Kuota ini membatasi berapa banyak properti yang dapat secara langsung bergantung pada satu properti, seperti yang didefinisikan dalam ekspresi rumus properti. Jumlah properti dependen harus lebih besar dari jumlah properti yang bergantung langsung untuk model aset. Minta peningkatan untuk kedua kuota jika ada properti yang lebih bergantung langsung daripada properti dependen untuk model aset.	20	Ya
Jumlah properti dependen dalam model aset	Kuota ini membatasi berapa banyak properti yang dapat secara langsung atau tidak langsung bergantung pada satu properti, seperti yang didefinisikan dalam ekspresi rumus properti.	30	Tidak

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah model komposit dalam model aset	Jumlah maksimum model komposit yang dapat Anda miliki pada model aset tunggal.	50	Ya
Kedalaman model komposit	Kedalaman maksimum pohon model komposit di setiap model aset, termasuk model inline dan component-model-based komposit.	2	Ya
Jumlah model aset unik yang menggunakan model komponen yang sama	Jumlah maksimum model aset unik yang memiliki setidaknya satu model component-model-based komposit yang secara langsung mereferensikan model aset tertentu dari tipe COMPONENT_MODEL.	20	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah variabel properti dalam ekspresi rumus properti	Misalnya, ada dua variabel properti, <code>power</code> dan <code>temp</code> , dalam ekspresi <code>avg(power) + max(temp)</code> . Ini juga berlaku untuk mengubah hasil perhitungan.	10	Tidak
Jumlah fungsi dalam ekspresi rumus properti	Misalnya, ada dua fungsi, <code>avg</code> dan <code>max</code> , dalam ekspresi <code>avg(power) + max(temp)</code> .	10	Tidak

Kuota untuk data properti AWS IoT SiteWise aset

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tingkat permintaan untuk operasi API data properti aset	Jumlah maksimum permintaan API data properti aset setiap detik yang dapat Anda lakukan Wilayah AWS di masing-masing detik Akun AWS. Kuota ini berlaku untuk operasi API seperti <code>GetAssetPropertyValue</code> dan <code>BatchPutAssetPropertyValue</code> .	1000	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah titik data setiap detik untuk setiap kualitas data untuk setiap properti aset	Kuota ini berlaku untuk jumlah maksimum titik data timestamp-quality-value (TQV) dengan stempel waktu yang sama dalam hitungan detik untuk setiap kualitas data untuk setiap properti aset. Anda dapat menyimpan hingga jumlah titik data berkualitas baik, tidak pasti, dan berkualitas buruk ini untuk setiap detik tertentu untuk setiap properti aset.	10	Tidak
Jumlah BatchPutAssetPropertyValues entri yang dicerna setiap detik di setiap properti aset untuk masing-masing Wilayah AWS untuk sebuah Akun AWS	Jumlah maksimum entri di setiap properti aset BatchPutAssetPropertyValues dari semua sumber, termasuk gateway SiteWise Edge, AWS IoT Core aturan, dan panggilan API.	10	Tidak

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tingkat titik data tertelan	Jumlah maksimum titik data timestamp-quality-value (TQV) yang dicerna per detik di masing-masing Wilayah AWS untuk a. Akun AWS	5000	Ya
Tarif permintaan untuk BatchGetAssetPropertyAggregates	Jumlah maksimum BatchGetAssetPropertyAggregates permintaan setiap detik yang dapat Anda lakukan Wilayah AWS di masing-masing permintaan Akun AWS.	200	Ya
Tarif permintaan untuk BatchGetAssetPropertyValue	Jumlah maksimum BatchGetAssetPropertyValue permintaan setiap detik yang dapat Anda lakukan Wilayah AWS di masing-masing permintaan Akun AWS.	500	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tarif permintaan untuk BatchGetAssetsPropertyHistory	Jumlah maksimum BatchGetAssetsPropertyHistory permintaan setiap detik yang dapat Anda lakukan.	200	Ya
Jumlah BatchPutAssetsPropertyHistory entri yang dicerna setiap detik untuk setiap properti aset untuk masing-masing Wilayah AWS dalam sebuah Akun AWS	Kuota ini berlaku untuk entri di setiap properti aset BatchPutAssetsPropertyHistory dari semua sumber, termasuk gateway SiteWise Edge, AWS IoT Core aturan, dan panggilan API.	10	Tidak
Tingkat GetAssetPropertyAggregates permintaan dan kueri BatchGetAssetsPropertyAggregates entri untuk setiap properti aset	Jumlah maksimum total GetAssetPropertyAggregates permintaan dan BatchGetAssetsPropertyAggregates entri untuk setiap properti aset per detik di masing-masing Wilayah AWS di masing-masing Akun AWS.	50	Tidak

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tingkat GetAssetProperty dan BatchGetAssetProperty untuk setiap properti aset	Jumlah maksimum total GetAssetProperty dan BatchGetAssetProperty untuk setiap properti aset setiap detik Wilayah AWS di masing-masing Akun AWS.	500	Tidak
Tingkat GetAssetPropertyHistory dan BatchGetAssetPropertyHistory untuk setiap properti aset	Jumlah maksimum total GetAssetPropertyHistory dan BatchGetAssetPropertyHistory untuk setiap properti aset setiap detik Wilayah AWS di masing-masing Akun AWS.	30	Tidak

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Permintaan tarif GetInterp olatedAss etPropert yValues	Jumlah maksimum GetInterp olatedAss etPropert yValues permintaa n setiap detik yang dapat Anda lakukan Wilayah AWS di masing-masing permintaan Akun AWS.	500	Ya
Jumlah hasil dalam setiap GetInterp olatedAss etPropert yValues permintaa n	Jumlah maksimum hasil yang akan dikembalikan untuk GetInterp olatedAss etPropert yValues permintaa n paginasi.	10	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tingkat titik data yang diambil dari GetAssetPropertyValueHistory dan BatchGetAssetPropertyHistory	Kecepatan byte maksimum (MB/detik) titik data diambil setiap detik untuk masing-masing di seberang dan. Wilayah AWS Akun AWS GetAssetPropertyValueHistory BatchGetAssetPropertyHistory Payload respons yang dievaluasi untuk kuota ini menggunakan bidang Timestamp-Quality-Value (TQV) untuk setiap titik data dan membulatkan ukuran byte untuk setiap permintaan API ke kenaikan 4KB berikutnya. Timestamp-quality-value (TQV) titik data yang diambil setiap detik bervariasi untuk setiap tipe data: Integer - hingga 5 Juta TQV per detik	100	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
	- Ganda - hingga 4 Juta TQV per detik - Boolean — hingga 6 Juta TQV per detik - String - bervariasi berdasarkan setiap ukuran nilai string.		

Kuota untuk gateway SiteWise Edge

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah gateway SiteWise Edge di masing-masing Wilayah AWS untuk Akun AWS	Jumlah maksimum gateway SiteWise Edge yang dapat Anda buat di Wilayah AWS untuk. Akun AWS	100	Ya
Jumlah sumber OPC UA di gateway SiteWise Edge	Jumlah maksimum sumber OPC UA yang dapat Anda konfigurasi di gateway SiteWise Edge.	100	Tidak
Jumlah total tujuan di gateway SiteWise Edge	Jumlah maksimum tujuan yang dapat Anda konfigurasi di gateway SiteWise Edge.	100	Tidak

Kuota untuk AWS IoT SiteWise Monitor

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah portal di masing-masing Wilayah AWS untuk Akun AWS	Jumlah maksimum portal SiteWise Monitor yang dapat Anda buat di Wilayah AWS for an Akun AWS.	100	Ya
Jumlah proyek di portal	Jumlah maksimum proyek yang dapat Anda buat dalam portal SiteWise Monitor.	100	Ya
Jumlah dasbor dalam sebuah proyek	Jumlah maksimum dasbor yang dapat Anda buat dalam proyek di SiteWise Monitor.	100	Ya
Jumlah aset root dalam suatu proyek	Jumlah maksimum aset tingkat atas yang dapat Anda tambahkan ke proyek di SiteWise Monitor.	1	Tidak
Jumlah visualisasi di dasbor	Jumlah maksimum elemen visual (seperti bagan, grafik, atau tabel) yang dapat Anda tambahkan ke dasbor di SiteWise Monitor.	10	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah metrik di setiap visualisasi dasbor	Jumlah maksimum metrik atau titik data yang dapat Anda tampilkan dalam satu visualisasi pada dasbor di SiteWise Monitor.	5	Ya
Jumlah ambang batas untuk setiap visualisasi dasbor	Jumlah maksimum level ambang batas yang dapat Anda atur untuk setiap visualisasi di dasbor di SiteWise Monitor.	12	Tidak

Kuota untuk impor AWS IoT SiteWise massal dan ekspor metadata

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah pekerjaan transfer metadata dalam antrian	Jumlah maksimum pekerjaan transfer PENDING metadata dalam antrian.	10	Ya
Ukuran file impor pekerjaan transfer metadata	Ukuran maksimum file yang diimpor (dalam MB).	100	Ya
Jumlah sumber daya AWS IoT SiteWise impor dalam suatu pekerjaan	Jumlah maksimum sumber daya AWS IoT SiteWise impor dalam satu pekerjaan . Sumber daya mencakup aset, dan model aset.	5000	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah sumber daya AWS IoT SiteWise ekspor dalam suatu pekerjaan	Jumlah maksimum sumber daya AWS IoT SiteWise ekspor dalam satu pekerjaan . Sumber daya mencakup aset, dan model aset.	5000	Ya

Kuota untuk impor data AWS IoT SiteWise massal

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Jumlah pekerjaan impor massal yang berjalan	Jumlah maksimum pekerjaan impor massal yang dapat dijalankan secara bersamaan.	100	Tidak
Ukuran file CSV	Ukuran file CSV maksimum (dalam GB) dalam pekerjaan impor massal.	10	Tidak
Ukuran file parquet yang tidak terkompresi	Ukuran file maksimum (dalam MB) untuk file parquet yang tidak terkompresi dalam pekerjaan impor massal.	270 MB	Tidak
Ukuran file CSV untuk konsumsi buffer	Ukuran file CSV maksimum (dalam MB) saat menggunakan penyerapan buffer	270 MB	Tidak

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
	pada pekerjaan impor massal.		
Ukuran kelompok baris parket yang tidak terkompresi	Ukuran maksimum grup baris parket yang tidak terkompresi.	64 MB	Tidak
Jumlah pengukuran unik dalam file parket	Jumlah maksimum pengukuran unik dalam file parket.	10000	Tidak
Jumlah hari antara stempel waktu di masa lalu dan hari ini untuk konsumsi buffer	Jumlah hari maksimum antara stempel waktu di masa lalu dan tanggal hari ini saat menggunakan konsumsi buffer.	30	Ya
Tarif permintaan untuk masing-masing CreateBulkImportJobs Wilayah AWS di masing-masing Akun AWS		10	Ya
Tarif permintaan ListBulkImportJobs untuk masing-masing Wilayah AWS di masing-masing Akun AWS		50	Ya

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tarif permintaan DescribeBulkImportJobs untuk masing-masing Wilayah AWS di masing-masing Akun AWS		50	Ya

Kuota untuk AWS IoT SiteWise Asisten API throttling

Kuota untuk batas pembatasan API AWS IoT SiteWise Asisten

Sumber Daya	Deskripsi	Kuota	Dapat Disesuaikan
Tingkat permintaan untuk InvokeAssistant operasi	Jumlah maksimum transaksi setiap menit (TPM) yang dapat dilakukan ke AWS IoT SiteWise InvokeAssistant API dalam sebuah Akun AWS. Batas TPM berlaku untuk semua wilayah yang didukung, dan dapat disesuaikan di beberapa wilayah.	10	Tidak

Kuota untuk deteksi anomali

Kuota untuk deteksi anomali dibagi antara dan AWS IoT SiteWise Amazon Lookout for Equipment. Untuk informasi selengkapnya, lihat [Kuota untuk menggunakan Lookout for Equipment](#).

Riwayat dokumen untuk Panduan AWS IoT SiteWise Pengguna

Tabel berikut menjelaskan dokumentasi untuk rilis ini AWS IoT SiteWise.

- Versi API: 2019-12-02

Perubahan	Deskripsi	Tanggal
Hubungkan aplikasi eksternal ke broker EMQX dengan nama pengguna dan kata sandi	Menambahkan dukungan untuk menghubungkan aplikasi eksternal menggunakan nama pengguna dan kata sandi ke broker AWS IoT Greengrass EMQX yang digunakan secara default saat membuat gateway V3 berkemampuan MQTT baru.	1 Mei 2025
Dukungan untuk gateway V3 berkemampuan MQTT di Edge SiteWise	Menambahkan fitur baru dan menghapus konten usang • Menambahkan dukungan untuk gateway V3 berkemampuan MQTT. Konfigurasi tujuan yang disempurnakan menggunakan filter jalur untuk berlangganan topik MQTT, termasuk konsumsi data real-time langsung ke AWS IoT SiteWise atau buffer konsumsi data menggunakan Amazon S3.	Februari 26, 2025

- Dirilis versi 3.0.0 dari kolektor IoT SiteWise OPC UA dan versi 4.0.0 dari komponen penerbit IoT untuk. SiteWise AWS IoT Greengrass V2
- Mengganti nama versi sebelumnya dari gateway SiteWise Edge yang dihosting sendiri menjadi aliran Klasik, gateway V2.
- Referensi dihapus ke AWS IoT Greengrass V1 dalam dokumentasi SiteWise Edge karena tidak lagi didukung untuk digunakan dengan AWS IoT SiteWise.

[Support untuk AWS IoT SiteWise Assistant](#)

Menambahkan dukungan untuk AWS IoT SiteWise Asisten - asisten bertenaga AI generatif.

November 18, 2024

[Menambahkan batas waktu sesi yang dapat dikonfigurasi untuk Edge SiteWise APIs](#)

Menambahkan pengaturan batas waktu sesi yang dapat dikonfigurasi untuk mengelola periode tidak aktif untuk AWS OpsHub dan Edge. SiteWise APIs

Oktober 31, 2024

[Menambahkan pengaturan proxy yang dapat dikonfigurasi untuk SiteWise Edge APIs](#)

Menambahkan mengelola informasi penyimpanan kepercayaan untuk mengaktifkan dukungan proxy HTTPS untuk gateway SiteWise Edge.

Oktober 31, 2024

Aktifkan CORS untuk Edge SiteWise APIs	Menambahkan dukungan CORS untuk SiteWise Edge APIs untuk mengaktifkan akses aplikasi web lintas domain yang aman.	September 30, 2024
Support untuk CloudRail dan sumber data Litmus Edge mitra	Menambahkan dukungan untuk keduanya CloudRail dan Litmus Edge sebagai sumber data mitra.	September 5, 2024
Ketersediaan umum untuk menjalankan SiteWise Edge di Siemens Industrial Edge	AWS IoT SiteWise sekarang mendukung ketersediaan umum menjalankan SiteWise Edge pada perangkat Siemens Industrial Edge.	Juli 24, 2024
Menambahkan dukungan untuk konfigurasi stempel waktu pada sumber data OPC UA	AWS IoT SiteWise sekarang mendukung konfigurasi stempel waktu untuk sumber data OPC UA.	Juli 24, 2024
Menambahkan dukungan untuk konversi tipe data pada sumber data OPC UA	AWS IoT SiteWise sekarang mendukung konversi tipe data untuk tipe data OPC UA yang tidak didukung.	Juli 24, 2024
Menambahkan dukungan untuk menjalankan pratinjau SiteWise Edge di Siemens Industrial Edge	AWS IoT SiteWise sekarang mendukung menjalankan pratinjau SiteWise Edge pada perangkat Siemens Industrial Edge.	26 November 2023

<u>Menambahkan dukungan untuk penyimpanan tingkat hangat</u>	AWS IoT SiteWise sekarang mendukung penyimpanan hangat, tingkat penyimpanan yang dikelola sepenuhnya yang memudahkan pelanggan untuk menyimpan dan mengakses data industri dengan aman.	15 November 2023
<u>Menambahkan dukungan untuk pengidentifikasi unik yang ditentukan pengguna</u>	AWS IoT SiteWise sekarang mendukung penggunaan pengidentifikasi unik yang ditentukan pengguna untuk aset, model aset, properti, dan hierarki.	15 November 2023
<u>Menambahkan dukungan untuk deteksi anomali multi variat aset industri</u>	AWS IoT SiteWise sekarang mendukung deteksi anomali multi variasi aset industri dengan integrasi data peralatan historis dan waktu nyata dengan Amazon Lookout for Equipment.	15 November 2023
<u>Menambahkan dukungan untuk konsumsi data deret waktu yang hemat biaya dan terukur di AWS IoT SiteWise</u>	AWS IoT SiteWise sekarang mendukung konsumsi data deret waktu yang efisien dan terukur yang diperlukan untuk kasus penggunaan analitis.	15 November 2023
<u>Ditambahkan dukungan untuk impor massal, ekspor, dan update</u>	AWS IoT SiteWise sekarang mendukung impor massal, ekspor, dan pembaruan metadata peralatan industri.	15 November 2023

Menambahkan dukungan untuk komponen model aset	AWS IoT SiteWise sekarang mendukung komponen model Aset untuk membantu pelanggan industri membuat komponen yang dapat digunakan kembali.	15 November 2023
Menambahkan dukungan untuk aplikasi dasbor IoT	AWS IoT SiteWise sekarang mendukung aplikasi dashboard open source di mana Anda dapat memvisualisasikan dan berinteraksi dengan data operasional.	15 November 2023
Memperbarui peran terkait layanan untuk AWS IoT SiteWise	AWS IoT SiteWise memiliki peran terkait layanan baru, dan dapat menjalankan kueri pencarian metadata, terhadap database. AWS IoT TwinMaker	6 November 2023
Penandaan yang diperbarui untuk sumber AWS IoT SiteWise daya aliran data	Menambahkan dukungan untuk menandai sumber daya aliran data.	18 Agustus 2022
Gerbang SiteWise Edge yang Diperbarui	Anda sekarang dapat mengonfigurasi penerbit untuk mengontrol data apa yang dikirim dari tepi ke cloud dan urutan yang dikirim ke cloud.	12 Januari 2022
Diperbarui AWS IoT SiteWise demo	Anda sekarang dapat menggunakan demo untuk membuat portal SiteWise Monitor.	10 Januari 2022

Manajemen penyimpanan yang diperbarui	Anda sekarang dapat menentukan periode retensi untuk mengontrol berapa lama data Anda disimpan di tingkat panas.	29 November 2021
Menambahkan dukungan untuk manajemen aliran data	Anda sekarang dapat mengambil data AWS IoT SiteWise sebelum Anda membuat model aset dan aset.	24 November 2021
Hirarki model aset yang diperbarui	Model aset anak sekarang dapat dikaitkan dengan beberapa model aset induk.	28 Oktober 2021
Peluncuran wilayah	Diluncurkan AWS IoT SiteWise di AWS GovCloud (AS-Barat).	29 September 2021
Fungsi yang diperbarui	Ditambahkan fitur berikut • Dalam metrik, Anda dapat menggunakan ekspresi bersarang dalam fungsi agregasi dan fungsi temporal. • Dalam transformasi, Anda dapat menggunakan fungsi pretrigger () untuk mengambil nilai variabel sebelum pembaruan properti yang memicu perhitungan transformasi saat ini.	Agustus 10, 2021
Interval waktu metrik khusus	Menambahkan dukungan untuk interval waktu khusus dan offset dalam metrik.	Agustus 3, 2021

Menggunakan AWS IoT SiteWise di tepi	Fitur pemrosesan tepi sekarang tersedia secara umum.	29 Juli 2021
Mengekspor data ke Amazon S3	AWS IoT SiteWise sekarang dapat mengekspor data ke Amazon S3.	27 Juli 2021
Titik akhir VPC ()AWS PrivateLink	Titik akhir VPC antarmuka untuk operasi API bidang kontrol sekarang tersedia secara umum.	15 Juli 2021
Mengubah	Transformasi sekarang dapat memasukkan beberapa variabel properti aset.	8 Juli 2021
Memperbarui fungsi timestamp ()	Dalam transformasi, Anda sekarang dapat memberikan variabel sebagai argumen untuk timestamp() fungsi.	16 Juni 2021
Alarm ketersediaan umum	Fitur alarm sekarang tersedia secara umum.	27 Mei 2021
Adaptor Protokol Modbus-TCP versi 2 dirilis	Versi 2 dari konektor Adaptor Protokol Modbus-TCP tersedia . Rilis ini menambahkan dukungan untuk ASCII, UTF8, dan string sumber yang ISO8859 dikodekan.	24 Mei 2021

Kuota layanan yang diperbarui

Menambahkan kuota berikut untuk [GetInterpolatedAssetPropertyValues](#) API: tingkat GetInterpolatedAssetPropertyValues permintaan, jumlah hasil per GetInterpolatedAssetPropertyValues permintaan, dan jumlah hari antara tanggal mulai di masa lalu dan hari ini untuk GetInterpolatedAssetPropertyValues .

29 April 2021

Ekspresi rumus yang diperbarui

Ditambahkan operator dan fungsi berikut:

22 April 2021

- Ditambahkan [operator](#) berikut: <, >, <=, >=, ==, !=, !, and, or, dan not.
- Ditambahkan [fungsi perbandingan](#) berikut: neq(x, y).
- Ditambahkan [fungsi string](#) berikut: join(), format(), dan f ' '.

Titik akhir VPC ()AWS PrivateLink

Menambahkan informasi tentang cara membuat koneksi pribadi antara virtual private cloud (VPC) Anda dan bidang AWS IoT SiteWise kontrol APIs dengan membuat titik akhir VPC antarmuka.

16 Maret 2021

Federasi IAM	Administrator dan pengguna portal SiteWise Monitor Anda sekarang dapat masuk ke portal yang ditetapkan dengan kredensial IAM mereka.	16 Maret 2021
Peluncuran wilayah	Diluncurkan AWS IoT SiteWise di Tiongkok (Beijing).	3 Februari 2021
SiteWise Konektor IoT versi 10 dirilis	Versi 10 dari SiteWise konektor IoT tersedia. Rilis ini dikonfigurasi StreamManager untuk meningkatkan penanganan ketika koneksi sumber terputus dan dibuat kembali. Versi ini juga menerima nilai OPC UA dengan ServerTimestamp ketika tidak SourceTimestamp tersedia.	22 Januari 2021
Fungsi tanggal dan waktu	AWS IoT SiteWise sekarang mendukung fungsi tanggal dan waktu.	21 Januari 2021
Sintaks fungsi	Anda sekarang dapat menggunakan Uniform Function Call Syntax (UFCS) untuk fungsi. AWS IoT SiteWise	11 Januari 2021
Integrasi dengan Grafana	Menambahkan informasi tentang cara memvisualisasikan AWS IoT SiteWise data di dasbor Grafana.	15 Desember 2020

[AWS IoT SiteWise rilis fitur](#)

15 Desember 2020

Anda sekarang dapat memantau data Anda dengan alarm, memproses data industri di edge, menggunakan Modbus TCP dan Ethernet/IP sumber ke gateway SiteWise Edge Anda, memfilter data yang masuk dengan deadband, dan banyak lagi.

- Menambahkan bagian [Pemantauan data dengan alarm](#) yang dapat Anda gunakan untuk menentukan, mengonfigurasi, dan merespons alarm. AWS IoT SiteWise
- Menambahkan bagian [pemrosesan Edge](#) yang dapat Anda gunakan untuk mengonfigurasi pemrosesan data industri Anda di perangkat edge Anda.
- Menambahkan bagian [Modbus TCP dan Ethernet/IP](#) ke dokumentasi sumber gateway Edge. SiteWise
- Menambahkan bagian [tujuan sumber](#) yang dapat Anda gunakan untuk menyesuaikan tempat Anda mengirim data industri yang masuk.
- Menambahkan bagian [pemfilteran OPC UA](#) yang dapat Anda gunakan untuk

mengontrol frekuensi dan jenis data yang dikirim ke gateway SiteWise Edge Anda dari server lokal industri Anda.

[AWS IoT SiteWise sekarang mendukung pelanggan yang dikelola CMKs.](#)

AWS IoT SiteWise sekarang mendukung enkripsi dengan pelanggan yang dikelola CMKs.

24 November 2020

[SiteWise Konektor IoT versi 8 dirilis](#)

Versi 8 dari SiteWise konektor IoT tersedia. Rilis ini meningkatkan stabilitas ketika konektor mengalami konektivitas jaringan intermiten.

19 November 2020

[Menggunakan string dan kondisional dalam ekspresi rumus](#)

Menambahkan informasi tentang cara string dan fungsi kondisional dalam ekspresi rumus untuk transformasi dan metrik.

16 November 2020

[Menelan data menggunakan pengelola AWS IoT Greengrass aliran](#)

Menambahkan informasi tentang cara menelan data IoT volume tinggi dari sumber data lokal menggunakan perangkat edge. AWS IoT Greengrass

16 September 2020

[Titik akhir VPC \(\)AWS PrivateLink](#)

Menambahkan informasi tentang cara membuat koneksi pribadi antara virtual private cloud (VPC) Anda dan AWS IoT SiteWise data APIs dengan membuat antarmuka VPC endpoint.

Selasa, 04 September 2020

SiteWise Konektor IoT versi 7 dirilis	Versi 7 dari SiteWise konektor IoT tersedia. Rilis ini memperbaiki masalah dengan metrik gateway SiteWise Edge.	14 Agustus 2020
Membuat pengguna IAM Identity Center dari konsol AWS IoT SiteWise	Menambahkan informasi tentang cara membuat pengguna IAM Identity Center di AWS IoT SiteWise konsol. Anda sekarang dapat membuat pengguna IAM Identity Center ketika Anda menetapkan pengguna ke portal baru atau yang sudah ada. Memperbarui tutorial Visualisasi dan berbagi data ladang angin untuk menggunakan fitur ini. Perubahan ini mengurangi jumlah langkah dalam tutorial.	4 Agustus 2020
Pemecahan masalah gateway SiteWise Edge yang ditingkatkan	Menambahkan informasi tambahan tentang cara memecahkan masalah gateway SiteWise Edge dan cara mengeksport sertifikat klien OPC UA untuk sumber .	18 Juni 2020
Dokumentasi tugas konsol	Menambahkan dokumentasi tugas konsol untuk Pemodelan aset industri , Kueri data properti aset , dan Berinteraksi dengan layanan lain . Anda dapat mengikuti petunjuk ini untuk menyelesaikan tugas di AWS IoT SiteWise konsol.	11 Juni 2020

Menganalisis tutorial data yang diekspor	Menambahkan tutorial yang dapat Anda ikuti untuk mempelajari cara menggunakan Amazon Athena untuk menganalisis data aset yang Anda ekspor ke Amazon S3 dengan templat fitur ekspor. AWS CloudFormation	27 Mei 2020
Peningkatan menggunakan ekspresi rumus	Menambahkan informasi rinci tentang perilaku properti AWS IoT SiteWise rumus dan menambahkan contoh cara menghitung titik data yang difilter.	18 Mei 2020
SiteWise Konektor IoT versi 6 dirilis	Versi 6 dari SiteWise konektor IoT tersedia. Rilis ini menambahkan dukungan untuk CloudWatch metrik dan penemuan otomatis tag OPC UA baru. Ini berarti Anda tidak perlu memulai ulang gateway SiteWise Edge Anda saat tag berubah untuk sumber OPC UA Anda. Versi konektor ini memerlukan manajer aliran dan perangkat lunak AWS IoT Greengrass Core v1.10.0 atau lebih tinggi.	29 April 2020

AWS IoT SiteWise rilis fitur

AWS IoT SiteWise rilis fitur. Sekarang Anda dapat mengelola gateway SiteWise Edge dengan API, menambahkan logo Anda ke portal, melihat metrik gateway SiteWise Edge, dan banyak lagi.

29 April 2020

- Menambahkan bagian [Mengekspor data ke Amazon](#) S3 dengan templat AWS CloudFormation yang dapat Anda gunakan untuk mengekspor nilai data baru ke bucket Amazon S3.
- Menambahkan bagian [Mengonfigurasi sumber data](#) yang meningkatkan dokumentasi sumber gateway SiteWise Edge dan menyertakan gateway SiteWise APIs Edge baru.
- Menambahkan bagian metrik gateway SiteWise Edge yang menjelaskan CloudWatch metrik yang diterbitkan oleh gateway SiteWise Edge.
- Menambahkan EC2 bagian Mengonfigurasi gateway SiteWise Edge di Amazon dengan AWS CloudFormation templat yang dapat Anda gunakan untuk mengonfigurasi dependens

i gateway SiteWise Edge dengan cepat pada instans Amazon. EC2

- Menambahkan bagian [peran layanan portal](#) yang menjelaskan fitur izin baru dari portal SiteWise Monitor.
- [Dokumentasi portal](#) yang diperbarui untuk peran layanan portal dan logo portal.
- Menambahkan bagian [Menandai AWS IoT SiteWise sumber daya Anda](#).
- Memperbarui bagian [Membuat dasbor \(CLI\)](#) untuk struktur definisi dasbor baru.
- Menambahkan bagian [Keamanan](#).

[Menelan data dari AWS IoT Events](#)

Menambahkan informasi tentang cara menelan data dari AWS IoT Events saat suatu peristiwa terjadi.

20 April 2020

[Memvisualisasikan dan berbagi data ladang angin di SiteWise Monitor tutorial](#)

Menambahkan tutorial yang dapat Anda ikuti untuk mempelajari cara menggunakan AWS IoT SiteWise Monitor untuk memvisualisasikan dan berbagi data aset.

12 Maret 2020

AWS IoT SiteWise konsep	Menambahkan glosarium AWS IoT SiteWise konsep yang dapat Anda gunakan untuk mempelajari tentang layanan dan istilah umum.	5 Maret 2020
Instruksi AWS IoT Greengrass instalasi dihapus	Menghapus instruksi penginstalan perangkat lunak AWS IoT Greengrass Inti dari Panduan AWS IoT SiteWise Pengguna. Panduan AWS IoT Greengrass Pengembangan menawarkan skrip pengaturan perangkat dan instruksi untuk disiapkan AWS IoT Greengrass di platform lain seperti Amazon EC2 dan Docker.	14 Februari 2020
Peningkatan menelan data menggunakan aturan AWS IoT Core	Menambahkan informasi terperinci tentang cara menggunakan dan cara memecahkan masalah tindakan AWS IoT SiteWise aturan, yang dapat Anda gunakan untuk menelan data dari pesan MQTT. AWS IoT Core	14 Februari 2020
SiteWise Konektor IoT versi 5 dirilis	Versi 5 dari SiteWise konektor IoT tersedia. Rilis ini memperbaiki masalah kompatibilitas dengan perangkat lunak AWS IoT Greengrass Core v1.9.4.	12 Februari 2020

[SiteWise Konektor IoT versi 4 dirilis](#)

Versi 4 dari SiteWise konektor IoT tersedia. Rilis ini memperbaiki masalah dengan koneksi ulang server OPC UA.

7 Februari 2020

[Aset industri pemodelan yang direstrukturasikan](#)

Merestrukturisasi bagian Memperbarui Aset dan Model menjadi beberapa topik dalam Pemodelan Aset Industri.

4 Februari 2020

- [Status aset dan model](#)
- [Mengelola aliran data untuk AWS IoT SiteWise](#)
- [Perbarui nilai atribut](#)
- [Mengasosiasikan dan memisahkan aset](#)
- [Perbarui aset dan model](#)
- [Hapus aset dan model di AWS IoT SiteWise](#)

[Menelan data dari AWS IoT hal-hal tutorial](#)

Menambahkan tutorial yang dapat Anda ikuti untuk mempelajari cara mengonfigurasi tindakan AWS IoT SiteWise aturan untuk menelan data dari armada baru atau yang sudah ada AWS IoT .

4 Februari 2020

[Merestrukturisasi mengambil data dari AWS IoT SiteWise](#)

[Merestrukturisasi bagian Mengambil Data menjadi dua bagian tingkat atas: Mengkueri nilai properti aset dan agregat dan Berinteraksi dengan layanan lain. AWS](#)

21 Januari 2020

Menerbitkan pembaruan nilai properti ke tutorial Amazon DynamoDB	Menambahkan tutorial yang dapat Anda ikuti untuk mempelajari cara menggunakan pemberitahuan nilai properti untuk menyimpan data aset di DynamoDB.	8 Januari 2020
Menggunakan ekspresi rumus	Menambahkan referensi ekspresi rumus untuk mengatur konstanta dan fungsi yang tersedia untuk digunakan dalam properti transformasi dan metrik. Properti Aset yang direstrukturisasi menjadi topik terpisah untuk setiap jenis properti.	Januari 7, 2020
Menggunakan filter node OPC UA	Menambahkan informasi tentang cara menggunakan filter node OPC UA untuk meningkatkan kinerja gateway SiteWise Edge saat menambahkan sumber gateway SiteWise Edge.	3 Januari 2020
Memutakhirkan konektor	Menambahkan informasi tentang cara meningkatkan gateway SiteWise Edge saat versi konektor baru dirilis.	Desember 30, 2019
SiteWise Konektor IoT versi 3 dirilis	Versi 3 dari SiteWise konektor IoT tersedia. Rilis ini menghapus persyaratan izin iot: *.	17 Desember 2019

[SiteWise Konektor IoT versi 2 dirilis](#)

Versi 2 dari SiteWise konektor IoT tersedia. Rilis ini menambahkan dukungan untuk beberapa sumber daya rahasia OPC UA.

10 Desember 2019

[Membuat dasbor \(\)AWS CLI](#)

Menambahkan informasi tentang cara membuat dasbor dalam AWS IoT SiteWise Monitor menggunakan AWS CLI.

6 Desember 2019

[AWS IoT SiteWise versi 2 dirilis](#)

Pratinjau yang dirilis untuk versi 2 dari AWS IoT SiteWise. Anda sekarang dapat menelan data melalui OPC UA, MQTT, dan HTTP, memodelkan data Anda dalam hierarki aset, dan memvisualisasikan data Anda dengan Monitor. SiteWise

2 Desember 2019

- Menulis ulang bagian [pemodelan aset](#) untuk perubahan aset, model aset, dan hierarki aset.
- Memperbarui bagian [konsumsi data untuk menyertakan langkah-langkah AWS IoT Greengrass s konektor dan bagian konsumsi](#) data non-gateway.
- Menambahkan [AWS IoT SiteWise Monitor](#) bagian dan [panduan aplikasi terpisah](#) yang menunjukkan cara menggunakan aplikasi web SiteWise Monitor.
- Ditambahkan [Data kueri dari AWS IoT SiteWise dan Berinteraksi dengan AWS layanan lain](#) bagian.
- Menulis ulang bagian [memulai](#) agar sesuai dengan pengalaman demo yang diperbarui.

[AWS IoT SiteWise versi 1 dirilis](#)

Merilis pratinjau awal untuk versi 1 dari AWS IoT SiteWise.

25 Februari 2019

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.