



Panduan Pengguna

AWS IoT Analytics



AWS IoT Analytics: Panduan Pengguna

Copyright © 2024 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 para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Apa itu AWS IoT Analytics?	1
Cara menggunakan AWS IoT Analytics	1
Fitur kunci	2
AWS IoT Analytics komponen dan konsep dan konsep	4
Akses AWS IoT Analytics	6
Kasus penggunaan	7
Memulai (konsol)	9
Masuk ke konsol AWS IoT Analytics.	10
Membuat Alur	10
Membuat Penyimpanan data	12
Membuat Alur	13
Membuat set data	15
Kirim data pesan dengan AWS IoT	17
Memeriksa kemajuan AWS IoT pesan	18
Akses hasil kueri	19
Jelajahi data Anda	19
Notebook template	21
Mulai	23
Membuat Alur	23
Membuat Penyimpanan Data	25
Kebijakan Amazon S3	25
Format file	27
Partisi kustom	30
Membuat Alur	33
Menelan data ke AWS IoT Analytics	34
Menggunakan broker AWS IoT pesan	35
Menggunakan BatchPutMessage API	38
Memantau data yang tertelan	39
Membuat Set Data	41
Kueri data	42
Mengakses data yang ditanyakan	42
Menjelajahi AWS IoT Analytics data	19
Amazon S3	43
AWS IoT Events	44

Amazon QuickSight	44
Notebook Jupyter	45
Menjaga beberapa versi dataset	45
Sintaks pesan	46
Bekerja dengan AWS IoT SiteWise data	47
Membuat set data	47
Isi set data	51
Tutorial: Kueri AWS IoT SiteWise data	53
Aktivitas ALUR	61
Aktivitas saluran	61
Aktivitas Datastore	61
AWS Lambda aktivitas	62
Contoh fungsi Lambda 1	63
Contoh fungsi Lambda 2	65
AddAttributes aktivitas	66
RemoveAttributes aktivitas	67
SelectAttributes aktivitas	68
Memfilter aktivitas	69
DeviceRegistryEnrich aktivitas	69
DeviceShadowEnrich aktivitas	71
Aktivitas Matematika	73
Operator aktivitas matematika dan fungsi	74
RunPipelineActivity	91
Pesan saluran	93
Parameter	93
Pesan saluran (konsol)	94
Memproses ulang pesan saluran (API)	95
Membatalkan aktivitas pemrosesan ulang saluran	95
Mengotomatisasi alur kerja	97
Kasus penggunaan	98
Menggunakan kontainer Docker	99
Variabel input/output kontainer Docker kustom	102
Izin	104
CreateDataset (Java dan AWS CLI)	106
Contoh 1 - membuat dataset SQL (java)	107
Contoh 2 - membuat dataset SQL dengan jendela delta (java)	108

Contoh 3 - membuat dataset kontainer dengan pemicu jadwal sendiri (java)	109
Contoh 4 - membuat dataset kontainer dengan dataset SQL sebagai pemicu (java)	110
Contoh 5 - membuat dataset SQL (CLI)	111
Contoh 6 - membuat dataset SQL dengan jendela delta (CLI)	111
Kontainerisasi notebook	113
Aktifkan kontainerisasi instance notebook yang tidak dibuat melalui AWS IoT Analytics konsol	113
Memperbarui ekstensi kontainerisasi notebook	116
Membuat citra kontainerisasi	116
Menggunakan wadah khusus	122
Memvisualisasi data	131
Visualisasi (konsol)	131
Visualisasi (QuickSight)	132
Penandaan	136
Dasar tanda	136
Menggunakan tanda dengan kebijakan IAM	137
Pembatasan tanda	139
Ekspresi SQL	141
Fungsi SQL yang didukung	142
Tipe data yang didukung	142
Fungsi yang didukung	143
Memecahkan masalah umum	144
Keamanan	145
AWS Identity and Access Management	145
Audiens	145
Mengautentikasi dengan identitas	146
Mengelola akses	150
Bekerja dengan IAM	152
Pencegahan wakil bingung lintas layanan	156
Contoh kebijakan IAM	162
Pemecahan masalah identitas dan akses	168
Pencatatan dan pemantauan	170
Alat pemantauan otomatis	170
Alat pemantauan manual	170
Pemantauan dengan CloudWatch Log	171
Pemantauan dengan CloudWatch Acara	176

Mencatat log panggilan API dengan CloudTrail	184
Validasi kepatuhan	189
Ketangguhan	190
Keamanan infrastruktur	190
Quotas	192
Perintah	193
Tindakan AWS IoT Analytics	193
AWS IoT Analyticsdata	193
Pemecahan masalah	194
Bagaimana cara saya tahu jika pesan saya masukAWS IoT Analytics?	194
Mengapa pipeline saya kehilangan pesan? Bagaimana cara cara saya memperbaikinya?	195
Mengapa tidak ada data di penyimpanan data saya?	196
Mengapa dataset saya hanya ditampilkan__dt?	196
Bagaimana cara membuat kode peristiwa yang didorong oleh penyelesaian kumpulan data? ..	197
Bagaimana cara mengkonfigurasi instance notebook saya dengan benar untuk digunakanAWS IoT Analytics?	197
Mengapa saya tidak bisa membuat notebook dalam sebuah instance?	197
Mengapa saya tidak melihat kumpulan data saya di Amazon QuickSight?	198
Mengapa saya tidak melihat tombol containerize pada Notebook Jupyter saya yang ada?	198
Mengapa instalasi plugin containerization saya gagal?	199
Mengapa plugin containerization saya melempar kesalahan?	199
Mengapa saya tidak melihat variabel saya selama penampung?	199
Variabel apa yang dapat saya tambahkan ke wadah saya sebagai input?	200
Bagaimana cara mengatur output kontainer saya sebagai masukan untuk analisis selanjutnya?	200
Mengapa dataset kontainer saya gagal?	200
Riwayat dokumen	201
Pembaruan sebelumnya	202
.....	cciii

Apakah AWS IoT Analytics itu?

AWS IoT Analytics mengotomatiskan langkah-langkah yang diperlukan untuk menganalisis data dari perangkat IoT. AWS IoT Analytics menyaring, mengubah, dan memperkaya data IoT sebelum menyimpannya di penyimpanan data deret waktu untuk analisis. Anda dapat mengatur layanan untuk mengumpulkan hanya data yang Anda butuhkan dari perangkat Anda, menerapkan transformasi matematika untuk memproses data, dan memperkaya data dengan metadata khusus perangkat seperti jenis perangkat dan lokasi sebelum menyimpannya. Kemudian, Anda dapat menganalisis data Anda dengan menjalankan kueri menggunakan mesin kueri SQL bawaan, atau melakukan analisis yang lebih kompleks dan inferensi machine learning. AWS IoT Analytics memungkinkan eksplorasi data lanjutan melalui integrasi dengan [Jupyter Notebook](#). AWS IoT Analytics juga memungkinkan visualisasi data melalui integrasi dengan [Amazon QuickSight](#). Amazon QuickSight tersedia di [Wilayah](#) berikut.

Analisis tradisional dan alat intelijen bisnis dirancang untuk memproses data terstruktur. Data IoT mentah sering kali berasal dari perangkat yang merekam data yang kurang terstruktur (seperti suhu, gerakan, atau suara). Akibatnya data dari perangkat ini dapat memiliki celah yang signifikan, pesan rusak, dan pembacaan palsu yang harus dibersihkan sebelum analisis dapat terjadi. Selain itu, data IoT seringkali hanya bermakna dalam konteks data lain dari sumber eksternal. AWS IoT Analytics memungkinkan Anda mengatasi masalah ini dan mengumpulkan sejumlah besar data perangkat, memproses pesan, dan menyimpannya. Anda kemudian dapat query data dan menganalisisnya. AWS IoT Analytics termasuk model pra-dibuat untuk kasus penggunaan IoT umum sehingga Anda dapat menjawab pertanyaan seperti perangkat mana yang akan gagal atau pelanggan mana yang berisiko meninggalkan perangkat yang dapat dikenakan mereka.

Cara menggunakan AWS IoT Analytics

Gambar berikut menunjukkan ikhtisar tentang bagaimana Anda dapat menggunakan AWS IoT Analytics.



Fitur kunci

Kumpulkan

- Terintegrasi dengan AWS IoT Core - AWS IoT Analytics sepenuhnya terintegrasi dengan AWS IoT Core sehingga dapat menerima pesan dari perangkat yang terhubung saat mereka streaming.
- Gunakan API batch untuk menambahkan data dari sumber apa pun - AWS IoT Analytics dapat menerima data dari sumber apa pun melalui HTTP. Itu berarti bahwa setiap perangkat atau layanan yang terhubung ke internet dapat mengirim data ke AWS IoT Analytics. Untuk informasi selengkapnya, lihat [BatchPutMessage](#) dalam Referensi API AWS IoT Analytics.
- Kumpulkan hanya data yang ingin Anda simpan dan analisis—Anda dapat menggunakan AWS IoT Analytics konsol AWS IoT Analytics untuk mengonfigurasi agar menerima pesan dari perangkat melalui filter topik MQTT dalam berbagai format dan frekuensi. AWS IoT Analytics memvalidasi bahwa data berada dalam parameter tertentu yang Anda tentukan dan buat saluran. Kemudian, layanan mengarahkan saluran ke jaringan pipa yang sesuai untuk pemrosesan pesan, transformasi, dan pengayaan.

Proses

- Membersihkan dan menyaring - AWS IoT Analytics memungkinkan Anda menentukan AWS Lambda fungsi yang dipicu ketika AWS IoT Analytics mendeteksi data yang hilang, sehingga Anda dapat menjalankan kode untuk memperkirakan dan mengisi kesenjangan. Anda juga dapat menentukan filter maksimum dan minimum serta ambang batas persentil untuk menghapus outliers dalam data Anda.

- Transform—AWS IoT Analytics dapat mengubah pesan menggunakan logika matematika atau kondisional Anda mendefinisikan, sehingga Anda dapat melakukan perhitungan umum seperti Celcius ke dalam konversi Fahrenheit.
- Enrich—AWS IoT Analytics Dapat memperkaya data dengan sumber data eksternal seperti ramalan cuaca, dan kemudian merutekan data ke penyimpanan AWS IoT Analytics data.

Menyimpan

- Penyimpanan data seri waktu -AWS IoT Analytics menyimpan data perangkat dalam penyimpanan data deret waktu yang dioptimalkan untuk pengambilan dan analisis yang lebih cepat. Anda juga dapat mengelola izin akses, menerapkan kebijakan penyimpanan data, dan mengekspor data Anda ke titik akses eksternal.
- Simpan data yang diproses dan mentah—AWS IoT Analytics Menyimpan data yang diproses dan juga secara otomatis menyimpan data mentah yang dicerna sehingga Anda dapat memprosesnya di lain waktu.

Menganalisis

- Jalankan query SQL Ad-hoc—AWS IoT Analytics menyediakan mesin kueri SQL sehingga Anda dapat menjalankan kueri ad-hoc dan mendapatkan hasil dengan cepat. Layanan ini memungkinkan Anda untuk menggunakan kueri SQL standar untuk mengekstrak data dari penyimpanan data untuk menjawab pertanyaan seperti jarak rata-rata yang ditempuh untuk armada kendaraan yang terhubung atau berapa banyak pintu di gedung pintar yang terkunci setelah jam 7 malam. Kueri ini dapat digunakan kembali bahkan jika perangkat yang terhubung, ukuran armada, dan persyaratan analitik berubah.
- Analisis deret waktu -AWS IoT Analytics mendukung analisis deret waktu sehingga Anda dapat menganalisis kinerja perangkat dari waktu ke waktu dan memahami bagaimana dan di mana perangkat digunakan, terus memantau data perangkat untuk memprediksi masalah pemeliharaan, dan memantau sensor untuk memprediksi dan bereaksi terhadap kondisi lingkungan.
- Notebook yang dihosting untuk analitik canggih dan pembelajaran mesin—AWS IoT Analytics mencakup dukungan untuk notebook yang di-host di Jupyter Notebook untuk analisis statistik dan pembelajaran mesin. Layanan ini mencakup satu set template notebook yang berisi model pembelajaran mesin AWS-authored dan visualisasi. Anda dapat menggunakan templat untuk memulai kasus penggunaan IoT yang terkait dengan pembuatan profil kegagalan perangkat, peramalan peristiwa seperti penggunaan rendah yang mungkin menandakan pelanggan akan meninggalkan produk, atau menyegmentasi perangkat berdasarkan tingkat penggunaan pelanggan (misalnya pengguna berat, pengguna akhir pekan) atau kesehatan perangkat.

Setelah Anda menulis buku catatan, Anda dapat mengemas dan menjalankannya pada jadwal yang Anda tentukan. Untuk informasi selengkapnya, lihat [Mengotomatisasi alur kerja Anda](#).

- **Prediksi**—Anda dapat melakukan klasifikasi statistik melalui metode yang disebut regresi logistik. Anda juga dapat menggunakan Memori Jangka Pendek Panjang (LSTM), yang merupakan teknik jaringan saraf yang kuat untuk memprediksi output atau keadaan proses yang bervariasi dari waktu ke waktu. Template notebook yang dibuat sebelumnya juga mendukung algoritma pengelompokan K-means untuk segmentasi perangkat, yang mengelompokkan perangkat Anda ke dalam kelompok perangkat sejenis. Template ini biasanya digunakan untuk memprofilkan kesehatan perangkat dan kondisi perangkat seperti unit HVAC di pabrik cokelat atau keausan bilah pada turbin angin. Sekali lagi, template notebook ini dapat terkandung dan dijalankan pada jadwal.

Bangun dan visualisasikan dan visualisasikan

- **QuickSight Integrasi Amazon-AWS IoT Analytics** menyediakan konektor ke Amazon QuickSight sehingga Anda dapat memvisualisasikan set data Anda di QuickSight dashboard.
- **Integrasi Konsol**—Anda juga dapat memvisualisasikan hasil atau analisis ad-hoc Anda di Notebook Jupyter yang disematkan di konsol AWS IoT Analytics '.

AWS IoT Analytics komponen dan konsep dan konsep

Channel

Saluran mengumpulkan data dari topik MQTT dan mengarsipkan pesan mentah dan belum diproses sebelum menerbitkan data ke alur. Anda juga dapat mengirim pesan ke saluran secara langsung menggunakan [BatchPutMessage](#) API. Pesan yang belum diproses disimpan di bucket Amazon Simple Storage Service (Amazon S3) yang Anda atau AWS IoT Analytics kelola.

Alur

Alur memakai pesan dari saluran dan memungkinkan Anda memproses pesan sebelum menyimpannya di penyimpanan data. Langkah-langkah pemrosesan, yang disebut aktivitas ([Aktivitas pipa](#)), melakukan transformasi pada pesan Anda seperti menghapus, mengganti nama, atau menambahkan atribut pesan, memfilter pesan berdasarkan nilai atribut, memanggil fungsi Lambda Anda pada pesan untuk pemrosesan lanjutan atau melakukan transformasi matematis untuk menormalkan data perangkat.

Penyimpanan data

Pipelines menyimpan pesan olahan mereka di penyimpanan data. Sebuah penyimpanan data bukan database, tetapi merupakan repositori scalable dan queryable dari pesan Anda. Anda dapat memiliki beberapa penyimpanan data untuk pesan yang berasal dari perangkat atau lokasi yang berbeda, atau difilter berdasarkan atribut pesan tergantung pada konfigurasi dan persyaratan pipeline Anda. Seperti pesan saluran yang belum diproses, pesan yang diproses penyimpanan data disimpan dalam bucket [Amazon S3](#) yang Anda atau AWS IoT Analytics kelola.

Set data data data ditetapkan?

Anda mengambil data dari penyimpanan data dengan membuat kumpulan data. AWS IoT Analytics memungkinkan Anda untuk membuat kumpulan data SQL atau kumpulan data kontainer.

Setelah Anda memiliki kumpulan data, Anda dapat menjelajahi dan mendapatkan wawasan tentang data Anda melalui integrasi menggunakan [Amazon QuickSight](#). Anda juga dapat melakukan fungsi analisis yang lebih canggih melalui integrasi dengan [Jupyter Notebook](#). Jupyter Notebook menyediakan alat ilmu data yang kuat yang dapat melakukan pembelajaran mesin dan berbagai analisis statistik. Untuk informasi selengkapnya, lihat [Template Notebook](#).

Anda dapat mengirim konten set data ke bucket [Amazon S3](#), memungkinkan integrasi dengan data lake yang ada atau akses dari aplikasi internal dan alat visualisasi. Anda juga dapat mengirim konten kumpulan data sebagai masukan ke [AWS IoT Events](#), layanan yang memungkinkan Anda untuk memantau perangkat atau proses untuk kegagalan atau perubahan dalam operasi, dan untuk memicu tindakan tambahan ketika peristiwa tersebut terjadi.

Set Data SQL SQL SQL SQL

Kumpulan data SQL mirip dengan tampilan terwujud dari database SQL. Anda dapat membuat set data SQL dengan menerapkan tindakan SQL. SQL data set dapat dihasilkan secara otomatis pada jadwal berulang dengan menentukan pemicu.

Set data kontainer kontainer kontainer

Kumpulan data kontainer memungkinkan Anda menjalankan alat analisis secara otomatis dan menghasilkan hasil. Untuk informasi selengkapnya, lihat [Mengotomatisasi alur kerja Anda](#). Ini menyatukan kumpulan data SQL sebagai input, wadah Docker dengan alat analisis Anda dan file perpustakaan yang dibutuhkan, variabel input dan output, dan pemicu jadwal opsional. Variabel input dan output memberi tahu gambar yang dapat dieksekusi di mana mendapatkan data dan menyimpan hasilnya. Pemicu dapat menjalankan analisis Anda ketika kumpulan data SQL selesai membuat kontennya atau sesuai dengan ekspresi jadwal waktu. Kumpulan data kontainer berjalan secara otomatis, menghasilkan, dan kemudian menyimpan hasil alat analisis.

Pemicu

Anda dapat secara otomatis membuat kumpulan data dengan menentukan pemicu. Pemicu dapat berupa interval waktu (misalnya, buat kumpulan data ini setiap dua jam) atau ketika konten kumpulan data lain telah dibuat (misalnya, buat kumpulan data ini saat `myOtherDataset` selesai membuat kontennya). Atau, Anda dapat menghasilkan konten kumpulan data secara manual dengan menggunakan [CreateDatasetContent](#) API.

Kontainer Docker

Anda dapat membuat container Docker sendiri untuk mengemas alat analisis Anda atau menggunakan opsi yang SageMaker disediakan. Untuk informasi selengkapnya, lihat [kontainer Docker](#). Anda dapat membuat container Docker Anda sendiri untuk mengemas alat analisis Anda atau menggunakan opsi yang disediakan oleh [SageMaker](#). Anda dapat menyimpan kontainer dalam registri [Amazon ECR](#) yang Anda tentukan sehingga tersedia untuk dipasang di platform yang Anda inginkan. Container Docker mampu menjalankan kode analitik kustom Anda yang disiapkan dengan Matlab, Octave, Wise.io, SPSS, R, Fortran, Python, Scala, Java, C ++, dan sebagainya. Untuk informasi selengkapnya, lihat [Kontainerisasi selengkapnya, lihat Kontainerisasi notebook](#).

Delta jendela

Jendela Delta adalah serangkaian interval waktu yang ditentukan pengguna, tidak tumpang tindih, dan bersebelahan. Delta jendela memungkinkan Anda untuk membuat isi set data dengan, dan melakukan analisis pada, data baru yang telah tiba di penyimpanan data sejak analisis terakhir. Anda membuat jendela delta dengan mengatur `deltaTime` di `filters` bagian `queryAction` dari kumpulan data. Untuk informasi selengkapnya, lihat API selengkapnya, lihat [CreateDataset](#) API. Biasanya, Anda ingin membuat konten kumpulan data secara otomatis dengan juga menyiapkan pemicu interval waktu (`triggers:schedule:expression`). Ini memungkinkan Anda memfilter pesan yang telah tiba selama jendela waktu tertentu, sehingga data yang terdapat dalam pesan dari jendela waktu sebelumnya tidak dihitung dua kali. Untuk informasi lebih lanjut, lihat [Contoh 6 -- membuat dataset SQL dengan jendela Delta \(CLI\)](#).

Akses AWS IoT Analytics

Sebagai bagian dari AWS IoT, AWS IoT Analytics menyediakan antarmuka berikut untuk memungkinkan perangkat Anda menghasilkan data dan aplikasi Anda untuk berinteraksi dengan data yang mereka hasilkan:

AWS Command Line Interface (AWS CLI)

Jalankan perintah untuk AWS IoT Analytics Windows, OS X, dan Linux. Perintah ini memungkinkan Anda untuk membuat dan mengelola hal-hal, sertifikat, aturan, dan kebijakan. Untuk mulai, lihat [Panduan Pengguna AWS Command Line Interface](#). Untuk informasi selengkapnya tentang perintah untuk AWS IoT, lihat [iot](#) di AWS Command Line Interface Referensi.

Important

Gunakan `aws iotanalytics` perintah untuk berinteraksi dengan AWS IoT Analytics.
Gunakan `aws iot` perintah untuk berinteraksi dengan bagian lain dari sistem IoT.

API AWS IoT

Bangun aplikasi IoT Anda menggunakan permintaan HTTP atau HTTPS. Tindakan API ini memungkinkan Anda untuk membuat dan mengelola hal-hal, sertifikat, aturan, dan kebijakan. Untuk informasi selengkapnya, lihat [Tindakan](#) di Referensi AWS IoT API.

AWS SDK

Buat AWS IoT Analytics aplikasi Anda menggunakan API khusus bahasa. SDK ini membungkus HTTP dan HTTPS API dan memungkinkan Anda untuk memprogram dalam salah satu bahasa yang didukung. Untuk informasi lebih lanjut, lihat [AWS SDK dan alat](#).

AWS IoT SDK Perangkat

Buat aplikasi yang berjalan di perangkat Anda yang mengirim pesan AWS IoT Analytics. Untuk informasi selengkapnya, lihat [SDK AWS IoT](#).

Konsol AWS IoT Analytics

Anda dapat membangun komponen untuk memvisualisasikan hasil di [AWS IoT Analytics konsol](#).

Kasus penggunaan

Perawatan prediktif prediktif

AWS IoT Analytics menyediakan template untuk membangun model pemeliharaan prediktif dan menerapkannya ke perangkat Anda. Misalnya, Anda dapat menggunakan AWS IoT Analytics untuk memprediksi kapan sistem pemanas dan pendingin cenderung gagal pada kendaraan

kargo yang terhubung sehingga kendaraan dapat dialihkan untuk mencegah kerusakan pengiriman. Atau, produsen auto dapat mendeteksi pelanggan mana yang telah memakai bantalan rem dan memperingatkan mereka untuk mencari perawatan untuk kendaraan mereka.

Pengisian kembali persediaan secara proaktif

AWS IoT Analytics memungkinkan Anda membangun aplikasi IoT yang dapat memantau inventaris secara real time. Misalnya, perusahaan makanan dan minuman dapat menganalisis data dari mesin penjual makanan dan secara proaktif menyusun ulang barang dagangan setiap kali pasokan hampir habis.

Penilaian efisiensi proses proses efisiensi proses proses penilaian efisiensi proses

Dengan AWS IoT Analytics, Anda dapat membangun aplikasi IoT yang terus-menerus memantau efisiensi proses yang berbeda dan mengambil tindakan untuk meningkatkan proses. Misalnya, perusahaan pertambangan dapat meningkatkan efisiensi truk bijihnya dengan memaksimalkan beban untuk setiap perjalanan. Dengan itu AWS IoT Analytics, perusahaan dapat mengidentifikasi beban yang paling efisien untuk lokasi atau truk dari waktu ke waktu, dan kemudian membandingkan penyimpangan dari beban target secara real time, dan lebih baik merencanakan pedoman terkemuka untuk meningkatkan efisiensi.

Pertanian Cerdas Cerdas Cerdas

AWS IoT Analytics dapat memperkaya data perangkat IoT dengan metadata kontekstual menggunakan data AWS IoT registri atau sumber data publik sehingga faktor analisis Anda dalam waktu, lokasi, suhu, ketinggian, dan kondisi lingkungan lainnya. Dengan analisis itu, Anda dapat menulis model yang menampilkan tindakan yang direkomendasikan untuk diambil perangkat Anda di lapangan. Misalnya, untuk menentukan kapan harus air, sistem irigasi dapat memperkaya data sensor kelembaban dengan data curah hujan, memungkinkan penggunaan air yang lebih efisien.

Memulai dengan AWS IoT Analytics (konsol)

Gunakan tutorial ini untuk membuat AWS IoT Analytics sumber daya (juga dikenal sebagai komponen) yang Anda butuhkan untuk menemukan wawasan berguna tentang data perangkat IoT Anda.

Catatan

- Jika Anda memasukkan karakter huruf besar dalam tutorial berikut, AWS IoT Analytics secara otomatis mengubahnya menjadi huruf kecil.
- AWS IoT Analytics Konsol memiliki fitur memulai satu klik untuk membuat saluran, pipeline, penyimpanan data, dan set data. Anda dapat menemukan fitur ini saat masuk ke AWS IoT Analytics konsol.
- Tutorial ini akan memandu Anda melalui setiap langkah untuk membuat AWS IoT Analytics sumber daya Anda.

Ikuti petunjuk di bawah ini untuk membuat AWS IoT Analytics saluran, pipeline, penyimpanan data, dan kumpulan data. Tutorial ini juga menunjukkan kepada Anda bagaimana menggunakan AWS IoT Core konsol untuk mengirim pesan yang akan dicerna AWS IoT Analytics.

Topik

- [Masuk ke konsol AWS IoT Analytics.](#)
- [Membuat Alur](#)
- [Membuat Penyimpanan data](#)
- [Membuat Alur](#)
- [Membuat set data](#)
- [Kirim data pesan dengan AWS IoT](#)
- [Memeriksa kemajuan AWS IoT pesan](#)
- [Akses hasil kueri](#)
- [Jelajahi data Anda](#)
- [Notebook template](#)

Masuk ke konsol AWS IoT Analytics.

Untuk memulai, Anda harus memiliki AWS akun. Jika Anda sudah memiliki AWS akun, buka <https://console.aws.amazon.com/iotanalytics/>.

Jika Anda tidak memiliki akun AWS, buat dengan mengikuti langkah-langkah ini.

Untuk membuat akun AWS

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

Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

Ketika 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 terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

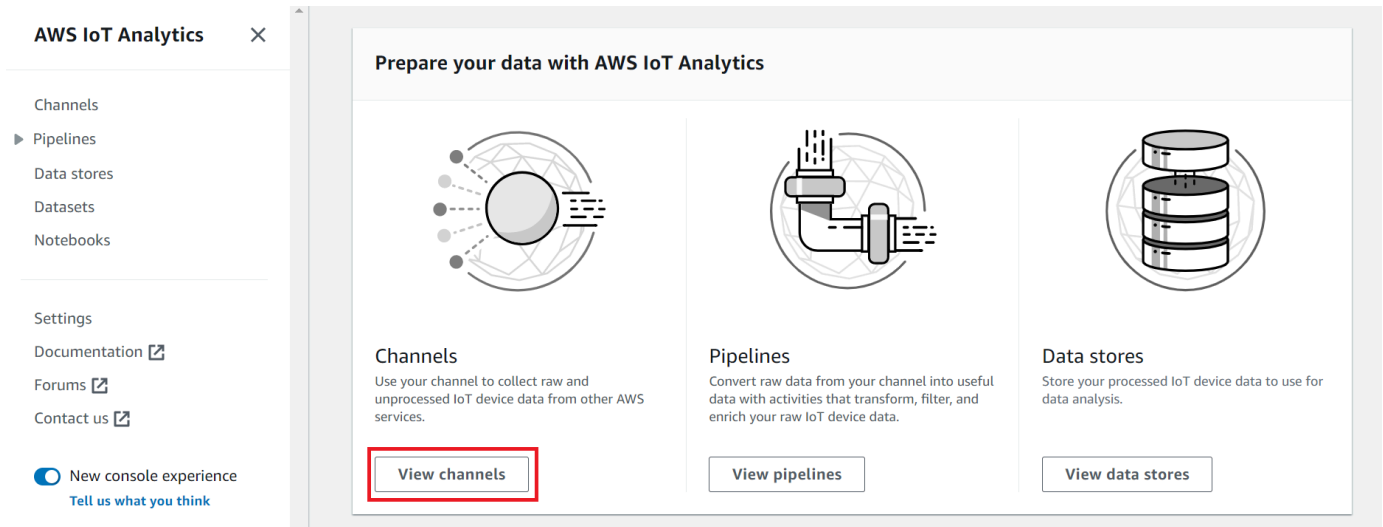
3. Masuk ke AWS Management Console dan arahkan ke <https://console.aws.amazon.com/iotanalytics/>.

Membuat Alur

Saluran mengumpulkan dan mengarsipkan data perangkat IoT mentah, belum diproses, dan tidak terstruktur. Ikuti langkah-langkah ini untuk membuat saluran Anda.

Untuk membuat saluran

1. Di <https://console.aws.amazon.com/iotanalytics/>, di AWS IoT Analytics bagian Siapkan data Anda dengan, pilih Lihat saluran.



Tip

Anda juga dapat memilih Saluran dari panel navigasi.

2. Pada halaman Saluran, pilih Buat saluran.
3. Pada halaman Tentukan detail saluran, masukkan detail tentang saluran Anda.
 - a. Masukkan nama saluran yang unik dan Anda dapat mengidentifikasi dengan mudah.
 - b. (Opsional) Untuk Tag, tambahkan satu tanda khusus atau lebih (pasangan nilai-kunci) ke saluran Anda. Tag dapat membantu Anda mengidentifikasi sumber daya yang Anda buat AWS IoT Analytics.
 - c. Pilih Selanjutnya.
4. AWS IoT Analytics menyimpan data perangkat IoT mentah dan belum diproses dalam bucket Amazon Simple Storage Service (Amazon S3). Anda dapat memilih bucket Amazon S3 Anda sendiri, yang dapat Anda akses dan kelola, atau AWS IoT Analytics dapat mengelola bucket Amazon S3 untuk Anda.
 - a. Dalam tutorial ini, untuk jenis Storage, pilih Service managed storage.
 - b. Untuk Pilih berapa lama untuk menyimpan data mentah Anda, pilih Tanpa batas.
 - c. Pilih Selanjutnya.
5. Pada halaman Konfigurasi sumber, masukkan informasi AWS IoT Analytics untuk mengumpulkan data pesan dari AWS IoT Core.

- a. Masukkan filter AWS IoT Core topik, misalnya, `update/environment/dht1`. Nanti dalam tutorial ini, Anda akan menggunakan filter topik ini untuk mengirim data pesan ke saluran Anda.
 - b. Di area peran IAM, pilih Buat baru. Di jendela Buat peran baru, masukkan nama untuk peran, lalu pilih Buat peran. Ini secara otomatis menciptakan peran dengan kebijakan yang sesuai yang melekat padanya.
 - c. Pilih Selanjutnya.
6. Tinjau pilihan Anda dan kemudian pilih Buat saluran.
 7. Verifikasi bahwa saluran baru Anda muncul di halaman Saluran.

Membuat Penyimpanan data

Toko data menerima dan menyimpan data pesan Anda. Sebuah penyimpanan data bukan database. Sebagai gantinya, penyimpanan data adalah repositori yang dapat diskalakan dan dapat dikueri dalam bucket Amazon S3. Anda dapat menggunakan beberapa penyimpanan data untuk pesan dari berbagai perangkat atau lokasi. Atau, Anda dapat memfilter data pesan tergantung pada konfigurasi dan persyaratan pipeline Anda.

Ikuti langkah-langkah ini untuk membuat penyimpanan data.

Untuk membuat penyimpanan data

1. Di <https://console.aws.amazon.com/iotanalytics/>, di AWS IoT Analytics bagian Siapkan data Anda dengan, pilih Lihat toko data.
2. Pada halaman Toko data, pilih Buat penyimpanan data.
3. Di halaman Tentukan data store, masukkan informasi dasar tentang penyimpanan data Anda.
 - a. Untuk ID penyimpanan data, masukkan ID penyimpanan data unik. Anda tidak dapat mengubah ID ini setelah Anda membuatnya.
 - b. (Opsional) Untuk Tag, pilih Tambahkan tag baru untuk menambahkan satu atau lebih tag khusus (pasangan kunci-nilai) ke penyimpanan data Anda. Tag dapat membantu Anda mengidentifikasi sumber daya yang Anda buat AWS IoT Analytics.
 - c. Pilih Selanjutnya.
4. Pada halaman Konfigurasi jenis penyimpanan, tentukan cara menyimpan data Anda.

- a. Untuk Jenis penyimpanan, pilih Penyimpanan terkelola layanan.
 - b. Untuk Mengkonfigurasi berapa lama Anda ingin menyimpan data yang diproses, pilih Tanpa batas.
 - c. Pilih Selanjutnya.
5. AWS IoT Analytics toko data mendukung format file JSON dan Parquet. Untuk format data penyimpanan data Anda, pilih JSON atau Parquet. Lihat [Format file](#) informasi selengkapnya tentang jenis file yang AWS IoT Analytics didukung.

Pilih Selanjutnya.

6. (Opsional) AWS IoT Analytics mendukung partisi khusus di penyimpanan data Anda sehingga Anda dapat melakukan kueri pada data yang dipangkas untuk meningkatkan latensi. Untuk informasi selengkapnya tentang partisi khusus yang didukung, lihat [Partisi kustom](#).

Pilih Selanjutnya.

7. Tinjau pilihan Anda dan kemudian pilih Buat penyimpanan data.
8. Verifikasi bahwa penyimpanan data baru Anda muncul di halaman Toko data.

Membuat Alur

Anda harus membuat pipeline untuk menghubungkan saluran ke penyimpanan data. Pipeline dasar hanya menentukan saluran yang mengumpulkan data dan mengidentifikasi penyimpanan data tempat pesan dikirim. Untuk informasi selengkapnya, lihat [Aktivitas Alur](#).

Untuk tutorial ini, Anda membuat pipeline yang hanya menghubungkan saluran ke penyimpanan data. Nantinya, Anda dapat menambahkan aktivitas pipeline untuk memproses data ini.

Ikuti langkah-langkah ini untuk membuat Alur.

Untuk membuat Alur

1. Di <https://console.aws.amazon.com/iotanalytics/>, di AWS IoT Analytics bagian Siapkan data Anda dengan, pilih Lihat saluran pipa.

Tip

Anda juga dapat memilih Alur dari panel navigasi.

2. Pada halaman Pipelines, pilih Create pipeline.
3. Masukkan rincian tentang Alur Anda.
 - a. Di Setup ID dan sumber pipa, masukkan nama pipeline.
 - b. Pilih sumber pipeline Anda, yang merupakan AWS IoT Analytics saluran yang akan dibaca oleh pipeline Anda.
 - c. Tentukan output pipeline Anda, yang merupakan penyimpanan data tempat data pesan yang diproses disimpan.
 - d. (Opsional) Untuk Tag, tambahkan satu tanda khusus atau lebih (pasangan nilai-kunci) ke Alur Anda.
 - e. Pada halaman atribut pesan Infer, masukkan nama atribut dan nilai contoh, pilih tipe data dari daftar, lalu pilih Tambahkan atribut.
 - f. Ulangi langkah sebelumnya untuk atribut sebanyak yang Anda butuhkan, lalu pilih Berikutnya.
 - g. Anda tidak akan menambahkan aktivitas pipeline apa pun sekarang. Pada halaman Enrich, transform, dan filter pesan, pilih Berikutnya.
4. Tinjau pilihan Anda dan kemudian pilih Buat Alur.
5. Verifikasi bahwa pipeline baru Anda muncul di halaman Pipelines.

Note

Anda membuat AWS IoT Analytics sumber daya sehingga mereka dapat melakukan hal berikut:

- Kumpulkan data pesan perangkat IoT mentah dan belum diproses dengan saluran.
- Simpan data pesan perangkat IoT Anda di penyimpanan data.
- Bersihkan, filter, ubah, dan per kaya data Anda dengan pipeline.

Selanjutnya, Anda akan membuat kumpulan data AWS IoT Analytics SQL untuk menemukan wawasan berguna tentang perangkat IoT Anda.

Membuat set data

Note

Dataset biasanya merupakan kumpulan data yang mungkin atau mungkin tidak diatur dalam bentuk tabel. Sebaliknya, AWS IoT Analytics buat kumpulan data Anda dengan menerapkan kueri SQL ke data di penyimpanan data Anda.

Anda sekarang memiliki saluran yang merutekan data pesan mentah ke pipeline yang menyimpan data di penyimpanan data yang dapat ditanyakan. Untuk query data, Anda membuat dataset. Sebuah dataset berisi pernyataan SQL dan ekspresi yang Anda gunakan untuk query penyimpanan data bersama dengan jadwal opsional yang mengulangi query pada hari dan waktu yang Anda tentukan. Anda dapat menggunakan ekspresi yang mirip dengan [ekspresi CloudWatch jadwal Amazon](#) untuk membuat jadwal opsional.


Untuk membuat set data

1. Di <https://console.aws.amazon.com/iotanalytics/>, di panel navigasi kiri, pilih Dataset.
2. Pada halaman Create dataset, pilih Create SQL.
3. Pada halaman Tentukan detail set data, tentukan detail kumpulan data Anda.
 - a. Masukkan nama untuk set data Anda.
 - b. Untuk Sumber penyimpanan data, pilih ID unik yang mengidentifikasi penyimpanan data yang Anda buat sebelumnya.
 - c. (Opsional) Untuk Tag, tambahkan satu tanda khusus atau lebih (pasangan nilai-kunci) ke set data Anda.
4. Gunakan ekspresi SQL untuk menanyakan data Anda dan menjawab pertanyaan analitis. Hasil kueri Anda disimpan dalam kumpulan data ini.
 - a. Di kolom kueri Penulis, masukkan kueri SQL yang menggunakan wildcard untuk menampilkan hingga lima baris data.

```
SELECT * FROM my_data_store LIMIT 5
```

Untuk informasi selengkapnya tentang fungsionalitas SQL yang didukung AWS IoT Analytics, lihat [Ekspresi SQL di AWS IoT Analytics](#).

- b. Anda dapat memilih Test query untuk memvalidasi bahwa input Anda benar dan menampilkan hasil dalam tabel mengikuti query.

 Note

- Pada titik dalam tutorial ini, data Anda mungkin kosong. Menjalankan query SQL pada datastore kosong tidak akan mengembalikan hasil, sehingga Anda mungkin hanya melihat __dt.
- Anda harus berhati-hati untuk membatasi kueri SQL Anda ke ukuran yang wajar sehingga tidak berjalan untuk waktu yang lama karena Athena [membatasi jumlah maksimum kueri yang berjalan](#). Karena itu, Anda harus berhati-hati untuk membatasi query SQL untuk ukuran yang wajar.

Kami menyarankan untuk menggunakanLIMIT klausa dalam kueri Anda selama pengujian. Setelah tes berhasil, Anda dapat menghapus klausa ini.

5. (Opsional) Saat Anda membuat konten kumpulan data menggunakan data dari kerangka waktu tertentu, beberapa data mungkin tidak tiba tepat waktu untuk diproses. Untuk memungkinkan penundaan, Anda dapat menentukan offset, atau delta. Untuk informasi selengkapnya, lihat [Mendapatkan pemberitahuan data terlambat melalui Amazon CloudWatch Events](#).

Anda tidak akan mengonfigurasi filter pemilihan data pada saat ini. Pada halaman Konfigurasi filter pemilihan data, pilih Berikutnya.

6. (Opsional) Anda dapat menjadwalkan kueri ini untuk dijalankan secara teratur untuk menyegarkan dataset. Jadwal set data dapat dibuat dan diedit kapan saja.

Anda tidak akan menjadwalkan kueri berulang pada saat ini, jadi pada halaman Atur jadwal kueri pilih Berikutnya.

7. AWS IoT Analytics akan membuat versi konten set data ini dan menyimpan hasil analisis Anda untuk periode yang ditentukan. Kami menyarankan 90 hari, namun Anda dapat memilih untuk menetapkan kebijakan retensi kustom Anda. Anda juga dapat membatasi jumlah versi yang disimpan dari konten kumpulan data Anda.

Anda dapat menggunakan periode retensi set data default sebagai Tanpa batas waktu dan menjaga Versi dinonaktifkan. Pada Konfigurasi hasil halaman analitik Anda, pilih Berikutnya.

8. (Opsional) Anda dapat mengonfigurasi aturan pengiriman hasil set data Anda ke tujuan tertentu, sepertiAWS IoT Events.

Anda tidak akan mengirimkan hasil Anda di tempat lain dalam tutorial ini, jadi pada halaman **Configure dataset content delivery rules**, pilih **Next**.

9. Tinjau pilihan Anda dan kemudian pilih **Buat set data**.
10. Verifikasi bahwa dataset baru Anda muncul di halaman **Dataset**.

Kirim data pesan dengan AWS IoT

Jika Anda memiliki saluran yang merutekan data ke pipeline, yang menyimpan data di penyimpanan data yang dapat ditanyakan, maka Anda siap untuk mengirim data perangkat IoT ke AWS IoT Analytics. Anda dapat mengirim data ke dalam AWS IoT Analytics dengan menggunakan opsi berikut:

- Gunakan broker AWS IoT pesan.
- Gunakan operasi AWS IoT Analytics [BatchPutMessageAPI](#).

Pada langkah-langkah berikut, Anda mengirim data AWS IoT pesan dari broker pesan di AWS IoT Core konsol sehingga AWS IoT Analytics dapat menelan data ini.

Note

Saat membuat nama topik untuk pesan Anda, perhatikan hal berikut:

- Nama-nama topik tidak peka huruf besar/kecil. Bidang bernama `example` dan `EXAMPLE` dalam payload yang sama dianggap duplikat.
- Nama topik tidak dapat dimulai dengan \$ karakter. Topik yang dimulai dengan \$ adalah topik yang dicadangkan dan hanya dapat digunakan oleh AWS IoT.
- Jangan sertakan informasi identitas pribadi dalam nama topik Anda karena informasi ini dapat muncul dalam komunikasi dan laporan yang tidak terenkripsi.
- AWS IoT Core tidak dapat mengirim pesan antar AWS akun atau AWS Wilayah.

Untuk mengirim data pesan dengan AWS IoT

1. Masuk ke [konsol AWS IoT](#) tersebut.
2. Di panel navigasi, pilih **Uji**, lalu pilih klien pengujian MQTT.
3. Pada halaman klien pengujian MQTT, pilih **Publikasikan ke topik**.

4. Untuk Nama topik, masukkan nama yang akan cocok dengan filter topik yang Anda masukkan saat membuat saluran. Contoh ini menggunakan `update/environment/dht1`.
5. Untuk payload Pesan, masukkan konten JSON berikut.

```
{
  "thingid": "dht1",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

6. (Opsional) Pilih Tambahkan Konfigurasi untuk opsi protokol pesan tambahan.
7. Pilih Terbitkan.

Ini menerbitkan pesan yang ditangkap oleh saluran Anda. Pipeline Anda kemudian merutekan pesan ke penyimpanan data Anda.

Memeriksa kemajuanAWS IoT pesan

Anda dapat memeriksa apakah pesan sedang dicerna ke saluran Anda dengan mengikuti langkah-langkah ini.

Untuk memeriksa kemajuanAWS IoT pesan

1. Masuk ke <https://console.aws.amazon.com/iotanalytics/>.
2. Di panel navigasi, pilih Saluran, lalu pilih nama saluran yang Anda buat sebelumnya.
3. Pada halaman detail Channel, gulir ke bawah ke bagian Monitoring, lalu sesuaikan kerangka waktu yang ditampilkan (1h 3h 12h 1d 3d 1w). Pilih nilai seperti 1w untuk melihat data selama seminggu terakhir.

Anda dapat menggunakan fitur serupa untuk memantau runtime aktivitas pipeline dan kesalahan pada halaman detail Pipeline. Dalam tutorial ini, Anda belum menentukan aktivitas sebagai bagian dari pipeline, jadi Anda seharusnya tidak melihat kesalahan runtime.

Untuk memantau aktivitas pipeline

1. Di panel navigasi, pilih Alur, lalu pilih nama Alur yang Anda buat sebelumnya.

2. Pada halaman detail Pipeline, gulir ke bawah ke bagian Monitoring, lalu sesuaikan kerangka waktu yang ditampilkan dengan memilih salah satu indikator kerangka waktu (1h 3h 12h 1d 3d 1w).

Akses hasil kueri

Konten set data adalah file yang berisi hasil kueri Anda, dalam format CSV.

1. Di <https://console.aws.amazon.com/iotanalytics/>, di panel navigasi kiri, pilih Dataset.
2. Pada halaman Dataset, pilih nama kumpulan data yang Anda buat sebelumnya.
3. Di halaman informasi set data, di sudut kanan atas, pilih Jalankan sekarang.
4. Untuk memeriksa apakah dataset sudah siap, lihat di bawah kumpulan data untuk pesan yang mirip dengan Anda telah berhasil memulai kueri untuk kumpulan data Anda. Tab konten Set data berisi hasil kueri dan tampilan Berhasil.
5. Untuk melihat pratinjau hasil kueri yang berhasil, pada tab Isi Set Data, pilih nama kueri. Untuk melihat atau menyimpan file CSV yang berisi hasil kueri, pilih Unduh.

Note

AWS IoT Analytics dapat menanamkan bagian HTML dari Notebook Jupyter pada halaman isi Dataset. Untuk informasi selengkapnya, lihat [Memvisualisasi AWS IoT Analytics data dengan konsol](#).

Jelajahi data Anda

Anda memiliki beberapa opsi untuk menyimpan, menganalisis, dan memvisualisasikan data Anda.

Amazon Simple Storage Service

Anda dapat mengirim konten set data ke bucket [Amazon S3](#), memungkinkan integrasi dengan data lake yang ada atau akses dari aplikasi internal dan alat visualisasi. Lihat bidang `contentDeliveryRules::destination::s3DestinationConfiguration` dalam [CreateDataset](#) operasi.

AWS IoT Events

Anda dapat mengirim konten set data sebagai input ke AWS IoT Events, layanan yang memungkinkan Anda memantau perangkat atau proses untuk kegagalan atau perubahan dalam operasi, dan untuk memulai tindakan tambahan ketika kejadian tersebut terjadi.

Untuk melakukan ini, buat kumpulan data menggunakan [CreateDataset](#) operasi dan tentukan AWS IoT Events input di bidang `contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. Anda juga harus menentukan `roleArn` peran, yang memberikan AWS IoT Analytics izin untuk dijalankan `iotevents:BatchPutMessage`. Setiap kali isi dataset dibuat, AWS IoT Analytics akan mengirim setiap entri konten dataset sebagai pesan ke AWS IoT Events input yang ditentukan. Misalnya, jika set data Anda berisi konten berikut.

```
"what", "who", "dt"  
"overflow", "sensor01", "2019-09-16 09:04:00.000"  
"overflow", "sensor02", "2019-09-16 09:07:00.000"  
"underflow", "sensor01", "2019-09-16 11:09:00.000"  
...
```

Kemudian AWS IoT Analytics mengirim pesan yang berisi bidang seperti berikut ini.

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

Anda akan ingin membuat AWS IoT Events input yang mengenali bidang yang Anda minati (satu atau lebih `what`, `who`, `dt`) dan untuk membuat model AWS IoT Events detektor yang menggunakan bidang input ini dalam peristiwa untuk memicu tindakan atau mengatur variabel internal.

Notebook Jupyter

[Jupyter Notebook](#) adalah solusi open source untuk menggunakan bahasa scripting untuk menjalankan eksplorasi data ad-hoc dan analisis lanjutan. Anda dapat menyelam lebih dalam dan menerapkan analisis yang lebih kompleks serta menggunakan metode pembelajaran mesin, seperti model pengelompokan dan regresi k-means untuk prediksi, pada data perangkat IoT Anda.

AWS IoT Analytics menggunakan instans SageMaker notebook Amazon untuk meng-host Notebook Jupyter. Sebelum membuat instance notebook, Anda harus membuat hubungan antara AWS IoT Analytics dan Amazon SageMaker:

1. Arahkan ke [SageMaker konsol](#) dan buat instance notebook:
 - a. Isi detailnya, lalu pilih Buat peran baru. Perhatikan ARN peran.
 - b. Membuat instans notebook.
2. Pergi ke [konsol IAM](#) dan memodifikasi SageMaker peran:
 - a. Buka perannya. Ini harus memiliki satu kebijakan yang dikelola.
 - b. Pilih Tambahkan kebijakan sebaris, lalu untuk Layanan, pilih IoTAnalytics. Pilih Pilih tindakan, lalu masukkan **GetDatasetContent** di kotak pencarian dan pilih. Pilih Tinjau Kebijakan.
 - c. Tinjau kebijakan untuk akurasi, masukkan nama, lalu pilih Buat kebijakan.

Ini memberikan izin peran yang baru dibuat untuk membaca dataset dari AWS IoT Analytics.

1. Kembali ke <https://console.aws.amazon.com/iotanalytics/>, dan di panel navigasi kiri, pilih Notebook. Pada halaman Notebook, pilih Buat buku catatan.
2. Pada halaman Pilih template, pilih template kosong IoT.
3. Pada halaman Set up notebook, masukkan nama untuk notebook Anda. Di Pilih sumber set data, pilih lalu pilih kumpulan data yang Anda buat sebelumnya. Di Pilih instance notebook, pilih instance notebook yang Anda buat SageMaker.
4. Setelah Anda meninjau pilihan Anda, pilih Buat Notebook.
5. Pada halaman Notebook, instance notebook Anda akan terbuka di SageMaker konsol [Amazon](#).

Notebook template

Template AWS IoT Analytics notebook berisi model pembelajaran mesin yang AWS ditulis dan visualisasi untuk membantu Anda memulai kasus AWS IoT Analytics penggunaan. Anda dapat menggunakan template notebook ini untuk mempelajari lebih lanjut atau menggunakannya kembali agar sesuai dengan data perangkat IoT Anda dan memberikan nilai langsung.

Anda dapat menemukan template notebook berikut di AWS IoT Analytics konsol:

- Mendeteksi anomali kontekstual - Penerapan deteksi anomali kontekstual dalam kecepatan angin yang diukur dengan model Poisson Exponentially Weighted Moving Average (PEWMA).
- Perkiraan keluaran panel surya - Penerapan model seri waktu piecewise, musiman, dan linier untuk memprediksi output panel surya.
- Perawatan prediktif pada mesin jet - Penerapan jaringan saraf Memori Jangka Pendek Panjang (LSTM) multivariat dan regresi logistik untuk memprediksi kegagalan mesin jet.
- Segmentasi pelanggan rumah pintar - Penerapan analisis k-means dan Principal Component Analysis (PCA) untuk mendeteksi segmen pelanggan yang berbeda dalam data penggunaan rumah pintar.
- Peramalan kemacetan kota pintar - Penerapan LSTM untuk memprediksi tingkat pemanfaatan jalan raya kota.
- Peramalan kualitas udara kota pintar - Penerapan LSTM untuk memprediksi polusi partikulat di pusat kota.

Memulai dengan AWS IoT Analytics

Bagian ini membahas perintah dasar yang Anda gunakan untuk mengumpulkan, menyimpan, memproses, dan melakukan kueri data Anda menggunakan AWS IoT Analytics. Contoh yang ditampilkan di sini menggunakan AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya tentang AWS CLI, lihat [Panduan Pengguna AWS Command Line Interface](#). Untuk informasi selengkapnya tentang perintah CLI yang tersedia AWS IoT, lihat [iot](#) di AWS Command Line Interface Referensi.

Important

Gunakan `aws iotanalytics` perintah untuk berinteraksi dengan AWS IoT Analytics menggunakan AWS CLI. Gunakan `aws iot` perintah untuk berinteraksi dengan bagian lain dari sistem IoT menggunakan AWS CLI.

Note

Ketahui saat Anda memasukkan nama AWS IoT Analytics entitas (saluran, set data, penyimpanan data, dan pipeline) dalam contoh berikut, bahwa setiap huruf besar yang Anda gunakan secara otomatis diubah menjadi huruf kecil oleh sistem. Nama-nama entitas harus dimulai dengan huruf kecil dan hanya berisi huruf kecil, garis bawah dan angka.

Membuat Alur

Saluran mengumpulkan dan mengarsipkan data pesan mentah dan belum diproses sebelum memublikasikan data ini ke pipeline. Pesan masuk dikirim ke saluran, jadi langkah pertama adalah membuat saluran untuk data Anda.

```
aws iotanalytics create-channel --channel-name mychannel
```

Jika ingin AWS IoT pesan dicerna AWS IoT Analytics, Anda dapat membuat AWS IoT aturan Mesin Aturan untuk mengirim pesan ke saluran ini. Hal ini ditunjukkan kemudian di [Menelan data ke AWS IoT Analytics](#). Cara lain untuk mendapatkan data ke saluran adalah dengan menggunakan AWS IoT Analytics perintah `BatchPutMessage`.

Untuk daftar saluran yang telah Anda buat:

```
aws iotanalytics list-channels
```

Untuk mendapatkan informasi lebih lanjut tentang saluran.

```
aws iotanalytics describe-channel --channel-name mychannel
```

Pesan saluran yang belum diproses disimpan dalam bucket Amazon S3 yang dikelola oleh AWS IoT Analytics, atau dalam satu yang dikelola oleh Anda. Gunakan `channelStorage` parameter untuk menentukan yang mana. Standarnya adalah bucket Amazon S3 yang dikelola layanan ini. Jika Anda memilih untuk menyimpan pesan saluran di bucket Amazon S3 yang Anda kelola, Anda harus memberikan AWS IoT Analytics izin untuk melakukan tindakan ini di bucket Amazon S3 atas nama Anda: `s3:GetBucketLocation` (verifikasi lokasi bucket), `s3:PutObject` (simpan), `s3:GetObject` (baca), `s3:ListBucket` (pemrosesan ulang).

Example

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-iot-analytics-bucket",
        "arn:aws:s3:::my-iot-analytics-bucket/*"
      ]
    }
  ]
}
```

Jika Anda membuat perubahan pada opsi atau izin penyimpanan saluran yang dikelola pelanggan, Anda mungkin perlu memproses ulang data saluran untuk memastikan bahwa data yang dicerna sebelumnya disertakan dalam konten set data. Lihat [Memproses ulang data saluran](#).

Membuat Penyimpanan Data

Toko data menerima dan menyimpan pesan Anda. Ini bukan database tetapi repositori scalable dan queryable dari pesan Anda. Anda dapat membuat beberapa penyimpanan data untuk menyimpan pesan yang berasal dari berbagai perangkat atau lokasi, atau Anda dapat menggunakan penyimpanan data tunggal untuk menerima semua AWS IoT pesan Anda.

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

Untuk daftar toko data yang telah Anda buat.

```
aws iotanalytics list-datastores
```

Untuk mendapatkan informasi lebih lanjut tentang penyimpanan data.

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

Kebijakan Amazon S3 untuk AWS IoT Analytics sumber daya

Anda dapat menyimpan pesan penyimpanan data yang diproses dalam bucket Amazon S3 yang dikelola oleh AWS IoT Analytics atau dalam salah satu yang Anda kelola. Saat Anda membuat penyimpanan data, pilih bucket Amazon S3 yang Anda inginkan dengan menggunakan `datastoreStorageParameter` API: Default bucket Amazon S3 yang dikelola layanan.

Jika Anda memilih pesan penyimpanan data disimpan dalam bucket Amazon S3 yang Anda kelola, Anda harus memberikan AWS IoT Analytics izin untuk melakukan tindakan ini di bucket Amazon S3 untuk Anda:

- `s3:GetBucketLocation`
- `s3:PutObject`
- `s3:DeleteObject`

Jika Anda menggunakan penyimpanan data sebagai sumber untuk kumpulan data kueri SQL, siapkan kebijakan bucket Amazon S3 yang memberikan AWS IoT Analytics izin untuk meminta permintaan Amazon Athena pada isi ember Anda.

Note

Kami menyarankan Anda menentukan `aws:SourceArn` dalam kebijakan ember Anda untuk membantu mencegah masalah keamanan wakil yang bingung. Ini membatasi akses dengan hanya mengizinkan permintaan yang berasal dari akun tertentu. Untuk informasi lebih lanjut tentang masalah deputi yang membingungkan, lihat [the section called "Pencegahan wakil bingung lintas layanan"](#).

Contoh berikut merupakan kebijakan bucket yang memberikan izin yang diperlukan ini.

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
```


Format file default adalah JSON. Anda hanya dapat menentukan satu format. Anda tidak dapat mengubah format file setelah membuat penyimpanan data.

`jsonConfiguration`

Berisi informasi konfigurasi dari format JSON.

`parquetConfiguration`

Berisi informasi konfigurasi dari format Parquet.

`schemaDefinition`

Informasi yang diperlukan untuk menentukan skema.

`columns`

Menentukan satu kolom atau lebih yang menyimpan data Anda.

Setiap skema dapat memiliki hingga 100 kolom. Setiap kolom dapat memiliki hingga 100 jenis nested.

`name`

Nama kolom.

Kendala panjang: 1-255 karakter.

`type`

Jenis data. Untuk informasi selengkapnya tentang tipe data yang didukung, lihat [Tipe data umum](#) di AWS Glue Panduan Pengembang.

Kendala panjang: 1-131072 karakter.

AWS IoT Analytics mendukung semua tipe data yang tercantum pada [Tipe Data di Amazon Athena](#) halaman, kecuali DECIMAL(*precision*, *scale*)-*precision*.

Membuat penyimpanan data (konsol)

Prosedur berikut menunjukkan cara membuat penyimpanan data yang menyimpan data dalam format Parquet.

Untuk membuat penyimpanan data

1. Masuk ke <https://console.aws.amazon.com/iotanalytics/>.

2. Di panel navigasi, pilih Penyimpanan data.
3. Pada Penyimpanan data halaman, pilih Membuat penyimpanan data.
4. Pada Memilih rincian penyimpanan data halaman, masukkan informasi dasar tentang penyimpanan data Anda.
 - a. Untuk ID penyimpanan data, masukkan ID penyimpanan data yang unik. Anda tidak dapat mengubah ID ini setelah Anda membuatnya.
 - b. (Opsional) Untuk Tag, pilih Tambahkan tag baru untuk menambahkan satu atau lebih tag kustom (pasangan kunci-nilai) ke penyimpanan data Anda. Tag dapat membantu Anda mengidentifikasi sumber daya yang Anda buat AWS IoT Analytics.
 - c. Pilih Selanjutnya.
5. Pada Mengkonfigurasi jenis penyimpanan halaman, tentukan cara menyimpan data Anda.
 - a. Untuk Jenis penyimpanan, pilih Penyimpanan terkelola layanan.
 - b. Untuk Konfigurasi berapa lama Anda ingin menyimpan data yang diproses, pilih Tanpa batas.
 - c. Pilih Selanjutnya.
6. Pada Mengkonfigurasi format data halaman, menentukan struktur dan format catatan data Anda.
 - a. Untuk Klasifikasi, pilih Parquet. Anda tidak dapat mengubah format ini setelah membuat penyimpanan data.
 - b. Untuk Sumber inferensi, pilih JSON tali untuk penyimpanan data Anda.
 - c. Untuk Tali, masukkan skema Anda dalam format JSON, seperti contoh berikut.


```
{
  "device_id": "0001",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

- d. Memilih Menyimpulkan skema.
- e. Di bawah Mengkonfigurasi skema Parquet, konfirmasi bahwa format sesuai dengan contoh JSON Anda. Jika formatnya tidak cocok, perbarui skema Parquet secara manual.
 - Jika Anda ingin skema Anda menampilkan lebih banyak kolom, pilih Tambahkan kolom baru, masukkan nama kolom, lalu pilih tipe data.

 Note

Secara default, Anda dapat memiliki 100 kolom untuk skema Anda. Untuk informasi selengkapnya, lihat [AWS IoT Analytics kuota](#).

- Anda dapat mengubah tipe data untuk kolom yang ada. Untuk informasi selengkapnya tentang tipe data yang didukung, lihat [Tipe data umum](#) di AWS Glue Panduan Pengembang.

 Note

Setelah membuat penyimpanan data, Anda tidak dapat mengubah tipe data untuk kolom yang ada.

- Untuk menghapus kolom yang ada, pilih [Menghapus kolom](#).

f. Pilih Selanjutnya.

7. (Opsional) AWS IoT Analytics mendukung partisi kustom di penyimpanan data Anda sehingga Anda dapat melakukan kueri pada data yang dipangkas untuk meningkatkan latensi. Untuk informasi selengkapnya tentang partisi kustom yang didukung, lihat [Partisi kustom](#).

Pilih Selanjutnya.

8. Pada [Memeriksa dan membuat halaman](#), tinjau pilihan Anda, lalu pilih [Membuat penyimpanan data](#).

 Important

Anda tidak dapat mengubah ID penyimpanan data, format file, atau tipe data untuk kolom setelah membuat penyimpanan data.

9. Verifikasi bahwa penyimpanan data baru Anda muncul di [Penyimpanan data](#) halaman.

Partisi kustom

AWS IoT Analytics mendukung partisi data sehingga Anda dapat mengatur data di penyimpanan data Anda. Bila Anda menggunakan partisi data untuk mengatur data, Anda dapat melakukan kueri pada

data yang dipangkas. Ini mengurangi jumlah data yang dipindai per permintaan dan meningkatkan latensi.

Anda dapat mempartisi data Anda sesuai dengan atribut data pesan atau atribut yang ditambahkan melalui aktivitas pipa.


Untuk memulai, aktifkan partisi data di penyimpanan data. Tentukan satu atau lebih dimensi partisi data dan hubungkan penyimpanan data yang dipartisi Anda keAWS IoT Analyticspipa. Kemudian, menulis kueri yang memanfaatkanWHEREklausul untuk mengoptimalkan kinerja.

Membuat penyimpanan data (konsol)

Prosedur berikut ini menunjukkan cara membuat penyimpanan data dengan partisi kustom.

Untuk membuat penyimpanan data

1. Masuk ke [konsol AWS IoT Analytics](#) tersebut.
2. Di panel navigasi, pilihPenyimpanan data.
3. PadaPenyimpanan datahalaman, pilihMembuat penyimpanan data.
4. PadaMenentukan rincian penyimpanan datahalaman, masukkan informasi dasar tentang penyimpanan data Anda.
 - a. UntukID penyimpanan data, masukkan ID penyimpanan data yang unik. Anda tidak dapat mengubah ID ini setelah Anda membuatnya.
 - b. (Opsional) UntukTag, pilihTambahkan tag baruuntuk menambahkan satu atau lebih tag kustom (pasangan kunci-nilai) ke penyimpanan data Anda. Tag dapat membantu Anda mengidentifikasi sumber daya yang Anda buatAWS IoT Analytics.
 - c. Pilih Selanjutnya.
5. PadaMengkonfigurasi jenis penyimpananhalaman, tentukan cara menyimpan data Anda.
 - a. UntukJenis penyimpanan, pilihPenyimpanan terkelola layanan.
 - b. UntukKonfigurasikan berapa lama Anda ingin menyimpan data yang diproses, pilihTanpa batas.
 - c. Pilih Selanjutnya.
6. PadaMengkonfigurasi format datahalaman, menentukan struktur dan format catatan data Anda.
 - a. Untuk format data penyimpanan data AndaKlasifikasi, pilihJSONatauParquet. Untuk informasi lebih lanjut tentangAWS IoT Analyticsjenis file yang didukung, lihat[Format file](#).


 Note

Anda tidak dapat mengubah format ini setelah membuat penyimpanan data.

- b. Pilih Selanjutnya.
7. Buat partisi khusus untuk penyimpanan data ini.
 - a. Untuk Tambahkan partisi data, pilih Aktifkan.
 - b. Untuk Sumber partisi data, tentukan informasi dasar tentang sumber partisi Anda.

Memilih Sumber sampel, dan pilih AWS IoT Analytics saluran yang mengumpulkan pesan untuk menyimpan data ini.

- c. Untuk Atribut sampel pesan, pilih atribut pesan yang ingin Anda gunakan untuk mempartisi penyimpanan data Anda. Kemudian, tambahkan pilihan Anda sebagai dimensi partisi atribut atau dimensi partisi timestamp di bawah Tindakan.

 Note

Anda hanya dapat menambahkan satu partisi timestamp ke penyimpanan data Anda.

- d. Untuk Dimensi partisi penyimpanan data khusus, tentukan informasi dasar tentang dimensi partisi Anda. Setiap atribut contoh pesan yang Anda pilih pada langkah sebelumnya akan menjadi dimensi partisi Anda. Sesuaikan setiap dimensi dengan opsi ini:
 - Jenis partisi- Tentukan jika dimensi partisi ini adalah Atribut atau Stempel waktu jenis partisi.
 - Nama atribut dan Nama dimensi- Secara default, AWS IoT Analytics akan menggunakan nama atribut sampel pesan yang Anda pilih sebagai pengenal untuk dimensi partisi atribut Anda. Edit nama atribut untuk menyesuaikan nama dimensi partisi Anda. Anda dapat menggunakan nama dimensi di WHERE klausa untuk mengoptimalkan kinerja query.
 - Nama dari setiap dimensi atribut partisi diawali dengan `__partition_`.
 - Untuk jenis partisi timestamp, AWS IoT Analytics menciptakan empat dimensi berikut dengan nama `__year`, `__month`, `__day`, `__hour`.
 - Memesan- Mengatur ulang dimensi partisi Anda untuk meningkatkan latensi untuk pertanyaan Anda.

Untuk Format stempel waktu, tentukan format partisi timestamp Anda dengan mencocokkan stempel waktu yang tertelan dari data pesan Anda. Anda dapat memilih salah satu AWS IoT Analytics pilihan format terdaftar, atau tentukan salah satu yang cocok dengan format data Anda. Pelajari selengkapnya tentang menentukan [Formatter tanggal](#).

Untuk menambahkan dimensi baru yang bukan atribut pesan, pilih Tambahkan partisi baru.

e. Pilih Selanjutnya.

8. Pada Memeriksa dan membuat halaman, tinjau pilihan Anda, lalu pilih Membuat penyimpanan data.

Important

- Anda tidak dapat mengubah ID penyimpanan data setelah membuat penyimpanan data.
- Untuk mengedit partisi yang ada, Anda harus membuat penyimpanan data lain dan memproses ulang data melalui pipa.

9. Verifikasi bahwa penyimpanan data baru Anda muncul di Penyimpanan data halaman.

Membuat Alur

Alur menggunakan pesan dari saluran dan memungkinkan Anda memproses dan memfilter pesan sebelum menyimpannya di penyimpanan data. Untuk menghubungkan saluran ke penyimpanan data, Anda membuat pipeline. Pipa yang paling sederhana mungkin tidak berisi aktivitas selain menentukan saluran yang mengumpulkan data dan mengidentifikasi penyimpanan data tempat pesan dikirim. Untuk informasi tentang jaringan pipa yang lebih rumit, lihat [Aktivitas pipa](#).

Saat memulai, kami sarankan Anda membuat pipeline yang tidak melakukan apa pun selain menghubungkan saluran ke penyimpanan data. Kemudian, setelah Anda memverifikasi bahwa data mentah mengalir ke penyimpanan data, Anda dapat memperkenalkan aktivitas pipeline tambahan untuk memproses data ini.

Jalankan perintah berikut ini untuk membuat Alur.

```
aws iotanalytics create-pipeline --cli-input-json file://mypipeline.json
```

`mypipeline.json` berisi konten berikut.

```
{
  "pipelineName": "mypipeline",
  "pipelineActivities": [
    {
      "channel": {
        "name": "mychannelactivity",
        "channelName": "mychannel",
        "next": "mystoreactivity"
      }
    },
    {
      "datastore": {
        "name": "mystoreactivity",
        "datastoreName": "mydatastore"
      }
    }
  ]
}
```

Jalankan perintah berikut ini untuk membuat daftar Alur Anda yang ada.

```
aws iotanalytics list-pipelines
```

Jalankan perintah berikut ini untuk menampilkan konfigurasi Alur.

```
aws iotanalytics describe-pipeline --pipeline-name mypipeline
```

Menelan data keAWS IoT Analytics

Jika Anda memiliki saluran yang merutekan data ke pipeline yang menyimpan data di penyimpanan data yang dapat ditanyakan, maka Anda siap untuk mengirim data pesanAWS IoT Analytics. Di sini kita menunjukkan dua metode untuk mendapatkan data keAWS IoT Analytics. Anda dapat mengirim pesan menggunakan brokerAWS IoT pesan atau menggunakanAWS IoT AnalyticsBatchPutMessage API.

Topik

- [Menggunakan brokerAWS IoT pesan](#)
- [Menggunakan BatchPutMessage API](#)

Menggunakan brokerAWS IoT pesan

Untuk menggunakan brokerAWS IoT pesan, Anda membuat aturan menggunakan mesinAWS IoT aturan. Aturan rute pesan dengan topik tertentu ke dalamAWS IoT Analytics. Tapi pertama-tama, aturan ini mengharuskan Anda untuk membuat peran yang memberikan izin yang diperlukan.

Membuat peran IAM

AgarAWS IoT pesan dialihkan keAWS IoT Analytics saluran, Anda menyiapkan aturan. Namun, Anda harus membuat peran IAM yang memberikan izin aturan tersebut untuk mengirim data pesan keAWS IoT Analytics saluran.

Jalankan perintah berikut untuk membuat peran.

```
aws iam create-role --role-name myAnalyticsRole --assume-role-policy-document file://arpd.json
```

Isi arpd.json file akan terlihat seperti berikut ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Kemudian, lampirkan dokumen kebijakan ke peran tersebut.

```
aws iam put-role-policy --role-name myAnalyticsRole --policy-name myAnalyticsPolicy --policy-document file://pd.json
```

Isi pd.json file akan terlihat seperti berikut ini.

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": "iotanalytics:BatchPutMessage",
        "Resource": [
          "arn:aws:iotanalytics:us-west-2:your-account-number:channel/mychannel"
        ]
      }
    ]
  }
}

```

MembuatAWS IoT aturan

BuatAWS IoT aturan yang mengirim pesan ke channel Anda.

```
aws iot create-topic-rule --rule-name analyticsTestRule --topic-rule-payload file://rule.json
```

Isirule.json file akan terlihat seperti berikut ini.

```

{
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [ {
    "iotAnalytics": {
      "channelName": "mychannel",
      "roleArn": "arn:aws:iam::your-account-number:role/myAnalyticsRole"
    }
  } ]
}

```

Gantiot/test dengan topik MQTT dari pesan yang harus dialihkan. Ganti nama saluran dan peran dengan yang Anda buat di bagian sebelumnya.

Mengirim pesan MQTTAWS IoT Analytics

Setelah Anda bergabung dengan aturan ke saluran, saluran ke pipeline, dan pipeline ke penyimpanan data, data apa pun yang cocok dengan aturan sekarang mengalirAWS IoT Analytics ke penyimpanan data yang siap ditanyakan. Untuk menguji ini, Anda dapat menggunakanAWS IoT konsol untuk mengirim pesan.

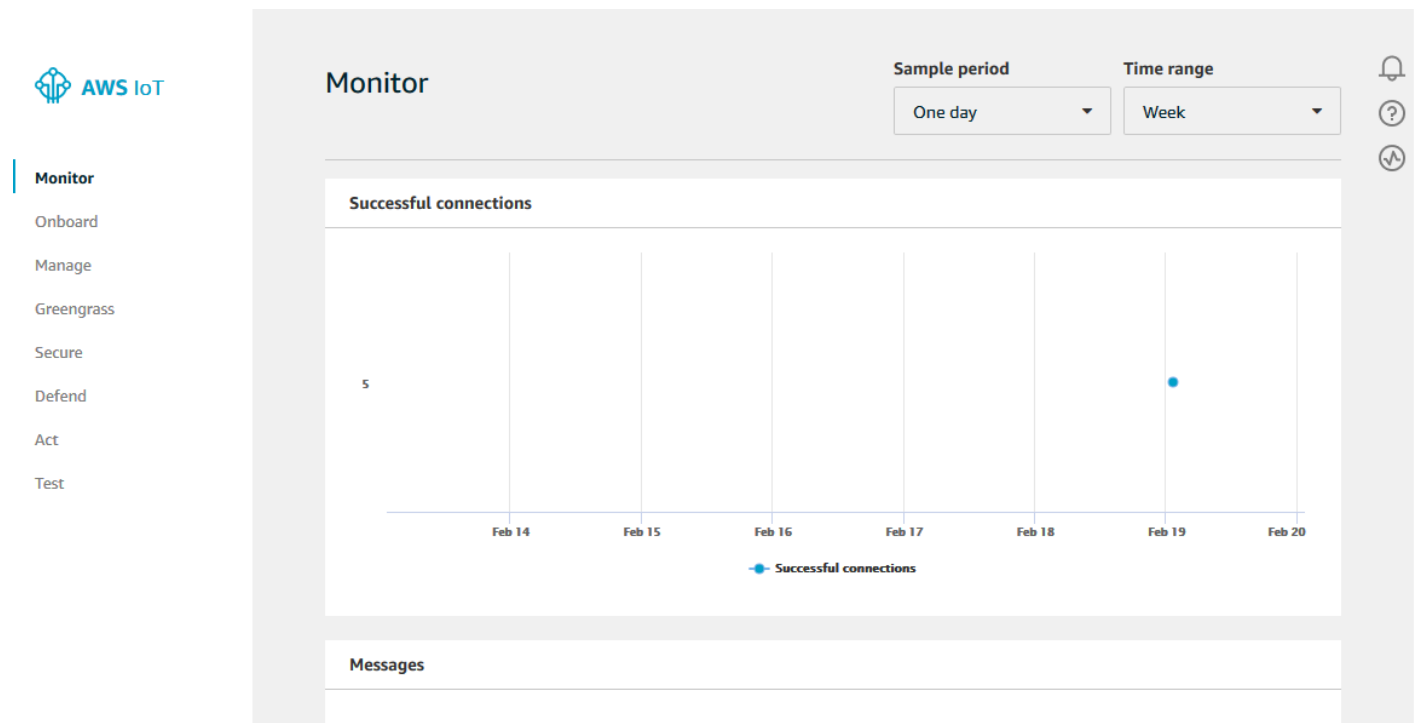
Note

Nama bidang payload pesan (data) yang Anda kirim ke AWS IoT Analytics.

- Harus hanya berisi karakter alfanumerik dan garis bawah (_); tidak ada karakter khusus lainnya yang diizinkan.
- Harus dimulai dengan karakter atau garis bawah (_).
- Tidak dapat berisi tanda hubung (-).
- Dalam istilah ekspresi reguler: `^[A-Za-z_]([A-Za-z0-9]* | [A-Za-z0-9][A-Za-z0-9_]*)$`.
- Tidak boleh lebih besar dari 255 karakter
- Tidak peka terhadap huruf besar/kecil. Bidang bernama `foo` dan `F00` dalam payload yang sama dianggap duplikat.

Misalnya, `{"temp_01": 29}` atau `{"_temp_01": 29}` valid, tetapi `{"temp-01": 29}`, `{"01_temp": 29}` atau `{"__temp_01": 29}` tidak valid dalam payload pesan.

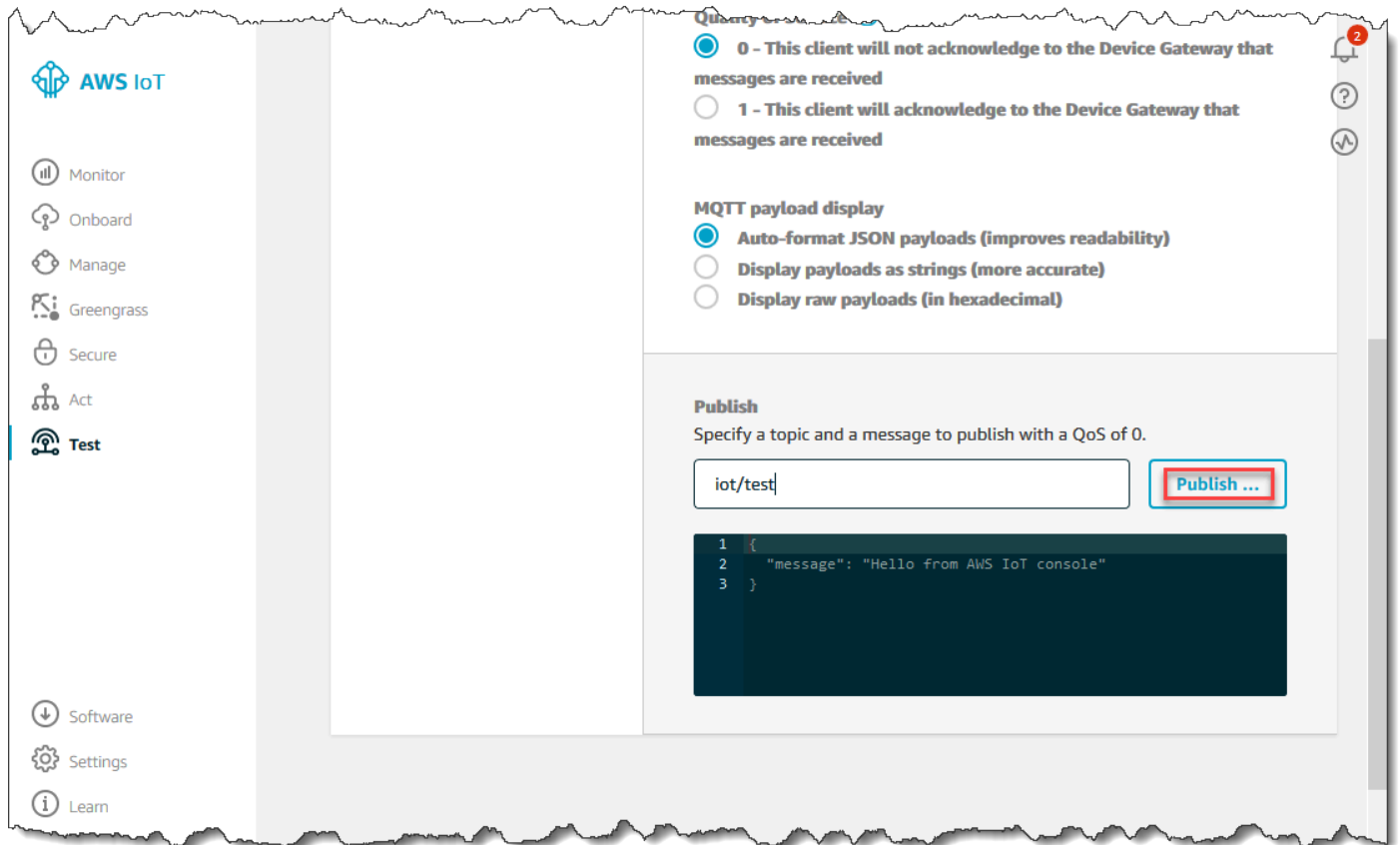
1. Di [AWS IoT konsol](#), di panel navigasi kiri, pilih Uji.



2. Pada halaman klien MQTT, di bagian Publikasikan, di Tentukan topik, ketik `iot/test`. Di bagian payload pesan, verifikasi konten JSON berikut ada, atau ketik jika tidak.

```
{
  "message": "Hello from the IoT console"
}
```

3. Pilih Terbitkan ke topik.



Ini menerbitkan pesan yang dirutekan ke penyimpanan data yang Anda buat sebelumnya.

Menggunakan BatchPutMessage API

Cara lain untuk memasukkan data pesan AWS IoT Analytics adalah dengan menggunakan perintah `BatchPutMessage` API. Metode ini tidak mengharuskan Anda menyiapkan AWS IoT aturan untuk merutekan pesan dengan topik tertentu ke saluran Anda. Tapi itu mengharuskan perangkat yang mengirimkan data/pesan ke saluran mampu menjalankan perangkat lunak yang dibuat dengan AWS SDK atau mampu menggunakan panggilan AWS CLI untuk `BatchPutMessage`.

1. Buat `filemessages.json` yang berisi pesan yang akan dikirim (dalam contoh ini hanya satu pesan yang dikirim).

```
[
  { "messageId": "message01", "payload": "{ \"message\": \"Hello from the CLI\n\" }" }
]
```

2. Jalankan perintah `batch-put-message`.

```
aws iotanalytics batch-put-message --channel-name mychannel --messages file://
messages.json --cli-binary-format raw-in-base64-out
```

Jika tidak ada kesalahan, Anda melihat output berikut ini.

```
{
  "batchPutMessageErrorEntries": []
}
```

Memantau data yang tertelan

Anda dapat memeriksa apakah pesan yang Anda kirim sedang dicerna ke saluran Anda dengan menggunakan AWS IoT Analytics konsol.

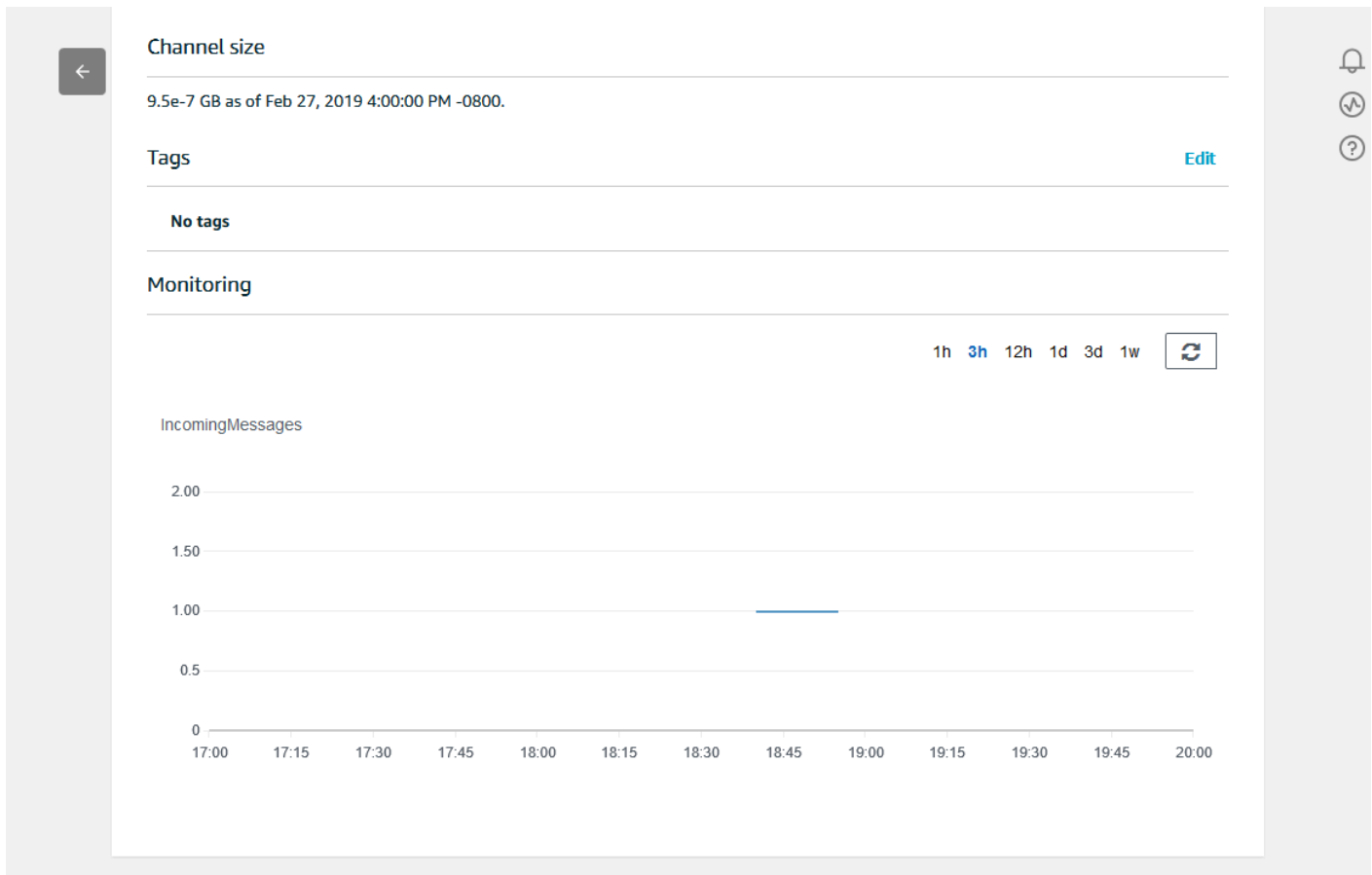
1. Di [AWS IoT Analytics konsol](#), di panel navigasi kiri, pilih Siapkan dan (jika perlu) pilih Saluran, lalu pilih nama saluran yang Anda buat sebelumnya.

The screenshot shows the AWS IoT Analytics console interface. On the left, a navigation sidebar lists 'Channels', 'Pipelines', 'Data stores', 'Data sets', and 'Notebooks', with 'Channels' selected. The main content area is titled 'Channels' and features a 'Create' button in the top right corner. Below the title is a table with the following data:

<input type="checkbox"/>	Name	Status	Created	Last updated
<input type="checkbox"/>	my_channel	ACTIVE	Sep 13, 2019 10:47:17 AM...	Sep 13, 2019 10:47:17 AM... ⋮

2. Pada halaman detail saluran, gulir ke bawah ke bagian Pemantauan. Sesuaikan kerangka waktu yang ditampilkan seperlunya dengan memilih salah satu indikator kerangka waktu (1h 3h 12h 1d

3d 1w). Anda akan melihat garis grafik yang menunjukkan jumlah pesan yang tertelan ke saluran ini selama jangka waktu yang ditentukan.



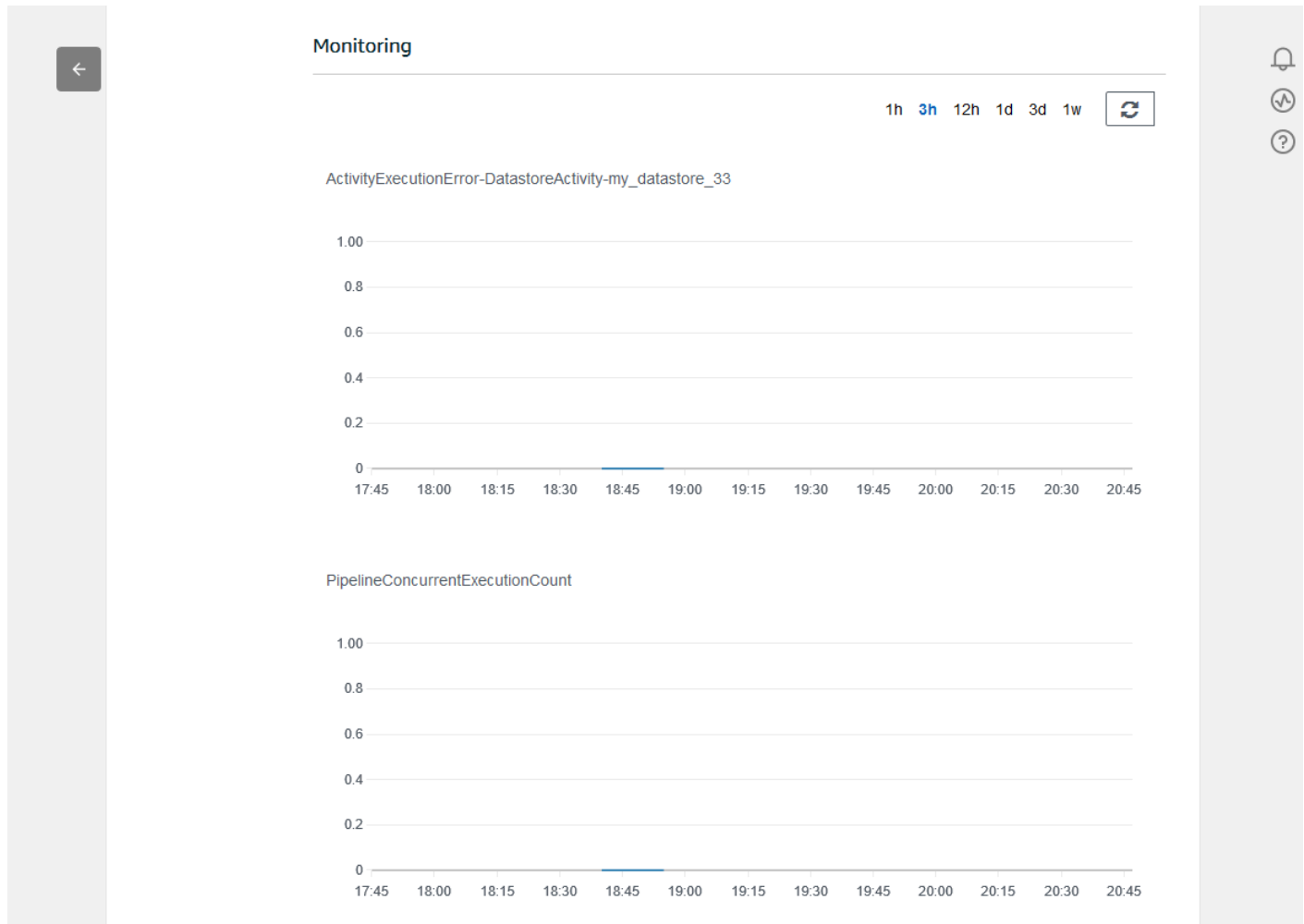
Kemampuan pemantauan serupa ada untuk memeriksa eksekusi aktivitas pipeline. Anda dapat memantau kesalahan eksekusi aktivitas di halaman detail pipeline. Jika Anda belum menentukan aktivitas sebagai bagian dari pipeline, maka 0 error eksekusi harus ditampilkan.

1. Di [AWS IoT Analytics konsol](#), di panel navigasi kiri, pilih Siapkan lalu pilih Pipelines, lalu pilih nama pipeline yang Anda buat sebelumnya.

Pipelines [Create](#)

<input type="checkbox"/> Name	Created	Last updated	
<input type="checkbox"/> my_pipeline	Sep 13, 2019 11:21:01 AM -0700	Sep 13, 2019 11:21:01 AM -0700	...

2. Pada halaman detail pipeline, gulir ke bawah ke bagian Monitoring. Sesuaikan kerangka waktu yang ditampilkan seperlunya dengan memilih salah satu indikator kerangka waktu (1h 3h 12h 1d 3d 1w). Anda akan melihat garis grafik yang menunjukkan jumlah kesalahan eksekusi aktivitas pipeline selama jangka waktu yang ditentukan.



Membuat Set Data

Anda mengambil data dari penyimpanan data dengan membuat dataset SQL atau dataset kontainer. AWS IoT Analytics dapat query data untuk menjawab pertanyaan analitis. Meskipun penyimpanan data bukan database, Anda menggunakan ekspresi SQL untuk query data dan menghasilkan hasil yang disimpan dalam dataset.

Topik

- [Kueri data](#)
- [Mengakses data yang ditanyakan](#)

Kueri data

Untuk query data, Anda membuat dataset. Sebuah dataset berisi SQL yang Anda gunakan untuk query penyimpanan data bersama dengan jadwal opsional yang mengulangi query pada hari dan waktu yang Anda pilih. Anda membuat jadwal opsional menggunakan ekspresi yang mirip dengan [ekspresi CloudWatch jadwal Amazon](#).

Jalankan perintah berikut ini untuk membuat set data.

```
aws iotanalytics create-dataset --cli-input-json file://mydataset.json
```

Dimana `mydataset.json` file berisi konten berikut.

```
{
  "datasetName": "mydataset",
  "actions": [
    {
      "actionName": "myaction",
      "queryAction": {
        "sqlQuery": "select * from mydatastore"
      }
    }
  ]
}
```

Jalankan perintah berikut ini untuk membuat konten set data dengan menjalankan kueri.

```
aws iotanalytics create-dataset-content --dataset-name mydataset
```

Tunggu beberapa menit sampai konten set data akan dibuat sebelum Anda melanjutkan.

Mengakses data yang ditanyakan

Hasil kueri adalah konten set data Anda, disimpan sebagai file, dalam format CSV. File ini tersedia untuk Anda melalui Amazon S3. Contoh berikut menunjukkan bagaimana Anda dapat memeriksa apakah hasil Anda siap dan mengunduh file tersebut.

Jalankan perintah `get-dataset-content` berikut.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```


Jika dataset Anda berisi data apapun, maka output dari `dataset-content`, memiliki `state`: `"SUCCEEDED"` distatus lapangan, seperti ini contoh berikut.

```
{
  "timestamp": 1508189965.746,
  "entries": [
    {
      "entryName": "someEntry",
      "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"
    }
  ],
  "status": {
    "state": "SUCCEEDED",
    "reason": "A useful comment."
  }
}
```

`dataURI` adalah URL ditandatangani untuk hasil output. Ini berlaku untuk waktu yang singkat (beberapa jam). Bergantung pada alur kerja Anda, Anda mungkin ingin selalu menelepon `dataset-content` sebelum mengakses konten karena memanggil perintah ini menghasilkan URL baru yang ditandatangani.

Menjelajahi AWS IoT Analytics data

Anda memiliki beberapa opsi untuk menyimpan, menganalisis, dan memvisualisasikan AWS IoT Analytics data Anda.

Topik di halaman ini:

- [Amazon S3](#)
- [AWS IoT Events](#)
- [Amazon QuickSight](#)
- [Notebook Jupyter](#)

Amazon S3

Anda dapat mengirim konten set data ke bucket [Amazon Simple Storage Service \(Amazon S3\)](#), yang memungkinkan integrasi dengan data lake

yang ada atau akses dari aplikasi internal dan alat visualisasi. Lihat lapangan `contentDeliveryRules::destination::s3DestinationConfiguration` di [CreateDataset](#).

AWS IoT Events

Anda dapat mengirim konten set data sebagai masukan AWS IoT Events, layanan yang memungkinkan Anda memantau perangkat atau proses untuk kegagalan atau perubahan dalam operasi, dan memicu tindakan tambahan ketika kejadian tersebut terjadi.

Untuk melakukan ini, buat kumpulan data menggunakan [CreateDataset](#) dan tentukan AWS IoT Events input di bidang `contentDeliveryRules::destination::iotEventsDestinationConfiguration::inputName`. Anda juga harus menentukan peran `roleArn` yang memberikan AWS IoT Analytics izin untuk mengeksekusi `"iotevents:BatchPutMessage"`. Setiap kali isi dataset dibuat, AWS IoT Analytics akan mengirim setiap entri konten dataset sebagai pesan ke AWS IoT Events input yang ditentukan. Misalnya, jika set data Anda berisi:

```
"what", "who", "dt"
"overflow", "sensor01", "2019-09-16 09:04:00.000"
"overflow", "sensor02", "2019-09-16 09:07:00.000"
"underflow", "sensor01", "2019-09-16 11:09:00.000"
...
```

kemudian AWS IoT Analytics akan mengirim pesan yang berisi bidang seperti ini:

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

dan Anda akan ingin membuat AWS IoT Events masukan yang mengenali bidang yang Anda minati (satu atau lebih `what`, `who`, `dt`) dan untuk membuat model AWS IoT Events detektor yang menggunakan bidang input ini dalam peristiwa untuk memicu tindakan atau mengatur variabel internal.

Amazon QuickSight

AWS IoT Analytics menyediakan integrasi langsung dengan [Amazon QuickSight](#). Amazon QuickSight adalah layanan analitik bisnis cepat yang dapat Anda gunakan untuk membangun visualisasi,

melakukan analisis ad-hoc, dan mendapatkan wawasan bisnis dari data Anda dengan cepat. Amazon QuickSight memungkinkan organisasi untuk menskalakan hingga ratusan ribu pengguna, dan memberikan kinerja responsif dengan menggunakan mesin dalam memori (SPICE) yang tangguh. Amazon QuickSight tersedia di [wilayah ini](#).

Notebook Jupyter

AWS IoT Analytics dataset juga dapat langsung dikonsumsi oleh Jupyter Notebook untuk melakukan analisis lanjutan dan eksplorasi data. Jupyter Notebook adalah solusi open source. Anda dapat menginstal dan mengunduh dari <http://jupyter.org/install.html>. Integrasi tambahan dengan SageMaker, solusi notebook yang dihosting Amazon, juga tersedia.

Menjaga beberapa versi dataset

Anda dapat memilih berapa banyak versi konten set data yang akan disimpan, dan untuk berapa lama, dengan menentukan nilai untuk `retentionPeriod` and `versioningConfiguration` bidang set data saat menjalankan [CreateDataset](#) dan [UpdateDataset](#) API:

```
...
"retentionPeriod": {
  "unlimited": "boolean",
  "numberOfDays": "integer"
},
"versioningConfiguration": {
  "unlimited": "boolean",
  "maxVersions": "integer"
},
...
```

Pengaturan kedua parameter ini bekerja sama untuk menentukan berapa banyak versi konten kumpulan data yang dipertahankan, dan untuk berapa lama, dengan cara berikut.

Retensi Periode	Retensi Periode:	Retensi Periode:
[tidak ditentukan]	unlimited = TRUE, numberOfDays = tidak diatur	tak terbatas = SALAH, numberOfDays = X

VersioningConfiguration: [tidak ditentukan]	Hanya versi terbaru ditambah versi terbaru yang berhasil (jika berbeda) yang dipertahankan selama 90 hari.	Hanya versi terbaru ditambah versi terbaru yang berhasil (jika berbeda) yang dipertahankan untuk waktu yang tidak terbatas.	Hanya versi terbaru ditambah versi terbaru yang berhasil (jika berbeda) yang dipertahankan selama X hari.
VersioningConfiguration: unlimited = TRUE, maxVersions tidak diatur	Semua versi dari 90 hari terakhir akan dipertahankan, terlepas dari berapa banyak.	Tidak ada batasan jumlah versi yang dipertahankan.	Semua versi dari hari X terakhir akan dipertahankan, terlepas dari berapa banyak.
VersioningConfiguration: tak terbatas = SALAH, maxVersions = Y	Tidak lebih dari versi Y dari 90 hari terakhir akan dipertahankan.	Hingga versi Y akan dipertahankan, terlepas dari berapa umurnya.	Tidak lebih dari versi Y dari hari X terakhir akan dipertahankan.

Sintaks pesan

Nama bidang payload pesan (data) yang Anda kirim ke AWS IoT Analytics:

- Harus hanya berisi karakter alfanumerik dan garis bawah (_); tidak ada karakter khusus lainnya yang diizinkan
- Harus dimulai dengan karakter atau garis bawah (_).
- Tidak dapat berisi tanda hubung (-).
- Dalam istilah ekspresi reguler: `"^[A-Za-z_]([A-Za-z0-9]* | [A-Za-z0-9][A-Za-z0-9_]*)$"`.
- Tidak boleh lebih besar dari 255 karakter.
- Tidak peka terhadap huruf besar/kecil. Bidang bernama "foo" dan "FOO" dalam payload yang sama dianggap duplikat.

Misalnya, {"temp_01": 29} atau {"_temp_01": 29} valid, tetapi {"temp-01": 29}, {"01_temp": 29} atau {"__temp_01": 29} tidak valid dalam payload pesan.

Bekerja dengan AWS IoT SiteWise data

AWS IoT SiteWise adalah layanan terkelola yang dapat Anda gunakan untuk mengumpulkan, menganalisis, dan memvisualisasikan data dari peralatan industri dalam skala besar. Layanan ini menyediakan kerangka pemodelan aset untuk membangun representasi perangkat, proses, dan fasilitas industri Anda.

Dengan AWS IoT SiteWise model aset, Anda dapat menentukan data peralatan industri apa yang akan dikonsumsi dan cara memproses data Anda menjadi metrik yang kompleks. Anda dapat mengkonfigurasi model aset untuk mengumpulkan dan memproses data di AWS Cloud. Untuk informasi selengkapnya, lihat [AWS IoT SiteWise Panduan Pengguna](#).

AWS IoT Analytics terintegrasi dengan AWS IoT SiteWise sehingga Anda dapat menjalankan dan menjadwalkan query SQL pada AWS IoT SiteWise data. Untuk memulai kueri AWS IoT SiteWise data, membuat penyimpanan data dengan mengikuti prosedur di [Konfigurasi pengaturan penyimpanan](#) di AWS IoT SiteWise Panduan Pengguna. Kemudian, ikuti langkah-langkah di [Membuat set data AWS IoT SiteWise data \(Konsol\)](#) atau di [Membuat set data AWS IoT SiteWise data AWS CLI](#) untuk membuat AWS IoT Analytics dataset dan menjalankan query SQL pada data industri Anda.

Topik

- [Membuat AWS IoT Analytics set data dengan AWS IoT SiteWise data](#)
- [Isi set data](#)
- [Tutorial: Kueri AWS IoT SiteWise data di AWS IoT Analytics](#)

Membuat AWS IoT Analytics set data dengan AWS IoT SiteWise data

Sesi AWS IoT Analytics dataset berisi pernyataan SQL dan ekspresi yang Anda gunakan untuk query data di penyimpanan data Anda bersama dengan jadwal opsional yang mengulangi query pada hari dan waktu yang Anda tentukan. Anda dapat menggunakan ekspresi yang mirip dengan [Amazon CloudWatch Ekspresi jadwal](#) untuk membuat jadwal opsional.

Note

Sebuah dataset biasanya kumpulan data yang mungkin atau mungkin tidak diatur dalam bentuk tabel. Sebaliknya, AWS IoT Analytics membuat set data Anda dengan menerapkan query SQL ke data Anda.

Ikuti langkah-langkah ini untuk memulai pembuatan set data AWS IoT SiteWise data.

Topik

- [Membuat set data AWS IoT SiteWise data \(Konsol\)](#)
- [Membuat set data AWS IoT SiteWise data AWS CLI](#)

Membuat set data AWS IoT SiteWise data (Konsol)

Gunakan langkah-langkah ini untuk membuat dataset di AWS IoT Analytics konsol untuk AWS IoT SiteWise data.


Membuat Set Data

1. Di <https://console.aws.amazon.com/iotanalytics/>, pada panel navigasi kiri, pilih Set Data.
2. Pada Buat set data halaman, pilih Membuat SQL.
3. Pada Tentukan set data halaman, tentukan rincian dataset Anda.
 - a. Masukkan nama untuk set data Anda.
 - b. Untuk Sumber penyimpanan data, pilih ID unik yang mengidentifikasi AWS IoT SiteWise penyimpanan data.
 - c. (Opsional) Untuk Tag, tambahkan satu tanda atau lebih (pasangan nilai-kunci) ke set data Anda.
4. Gunakan ekspresi SQL untuk query data Anda dan menjawab pertanyaan analitis.
 - a. Di Kueri penulis bidang, masukkan query SQL yang menggunakan wildcard untuk menampilkan hingga lima baris data.

```
SELECT * FROM my_iotsitewise_datastore.asset_metadata LIMIT 5
```

Untuk informasi selengkapnya tentang fungsionalitas SQL yang didukung AWS IoT Analytics, lihat [Ekspresi SQL di AWS IoT Analytics](#). Atau, lihat [Tutorial: Kueri AWS IoT SiteWise data di AWS IoT Analytics](#) untuk contoh kueri statistik yang dapat memberikan wawasan data Anda.

- b. Anda dapat memilih kueri uji untuk memvalidasi bahwa input Anda benar, dan untuk menampilkan hasil dalam tabel berikut query.


 Note

Karena Amazon Athena [membatasi jumlah maksimum kueri yang berjalan](#), Anda harus membatasi kueri SQL Anda ke ukuran yang wajar sehingga tidak berjalan untuk jangka waktu yang lama.

5. (Opsional) Bila Anda membuat isi dataset menggunakan data dari kerangka waktu yang ditentukan, beberapa data mungkin tidak tiba tepat waktu untuk diproses. Untuk memungkinkan penundaan, Anda dapat menentukan offset, atau delta. Untuk informasi selengkapnya, lihat [Mendapatkan pemberitahuan data terlambat melalui Amazon CloudWatch Events](#).

Setelah mengkonfigurasi filter pemilihan data pada [Mengkonfigurasi filter pemilihan data](#) halaman, pilih [Selanjutnya](#).

6. (Opsional) Pada [Mengatur halaman jadwal kueri](#), Anda dapat menjadwalkan kueri ini untuk menjalankan secara teratur untuk menyegarkan dataset. Jadwal set data dapat dibuat dan diedit kapan saja.

 Note

Data AWS IoT SiteWise menelan AWS IoT Analytics setiap enam jam. Sebaiknya pilih frekuensi enam jam atau lebih.

Pilih dan pilih untuk [Frekuensi](#) dan kemudian pilih [Selanjutnya](#).

7. AWS IoT Analytics akan membuat versi konten dataset ini dan menyimpan hasil analisis Anda untuk periode yang ditentukan. Sebaiknya 90 hari, namun Anda dapat memilih untuk menetapkan kebijakan penyimpanan kustom Anda. Anda juga dapat membatasi jumlah versi yang tersimpan dari konten dataset Anda.

Setelah memilih opsi Anda di [Konfigurasi hasil dataset](#) halaman, pilih [Selanjutnya](#).

8. (Opsional) Anda dapat mengonfigurasi aturan pengiriman hasil dataset Anda ke tujuan tertentu, seperti AWS IoT Events.

Setelah memilih opsi Anda di **Mengonfigurasi aturan pengiriman konten dataset halaman**, pilih **Selanjutnya**.

9. Tinjau pilihan Anda dan kemudian pilih **Buat set data**.
10. Verifikasi bahwa dataset baru Anda muncul di **Set Data halaman**.

Membuat set data AWS IoT SiteWise data AWS CLI)

Jalankan AWS CLI perintah untuk memulai query AWS IoT SiteWise data.

Contoh yang ditunjukkan di sini menggunakan AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya tentang AWS CLI, lihat [Panduan Pengguna AWS Command Line Interface](#). Untuk informasi selengkapnya tentang perintah CLI AWS IoT Analytics, lihat [iotanalytics](#) di AWS Command Line Interface Referensi.

Membuat Set Data

1. Jalankan `create-dataset` perintah untuk membuat set data.

```
aws iotanalytics create-dataset --cli-input-json file://my_dataset.json
```

Di mana `my_dataset.json` berisi konten berikut.

```
{
  "datasetName": "my_dataset",
  "actions": [
    {
      "actionName": "my_action",
      "queryAction": {
        "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
LIMIT 5"
      }
    }
  ]
}
```


Untuk informasi selengkapnya tentang fungsionalitas SQL yang didukung AWS IoT Analytics, lihat [Ekspresi SQL di AWS IoT Analytics](#). Atau, lihat [Tutorial: Kueri AWS IoT SiteWise data di AWS IoT Analytics](#) untuk contoh kueri statistik yang dapat memberikan wawasan data Anda.

2. Jalankan `create-dataset-content` perintah untuk membuat konten dataset Anda dengan menjalankan kueri Anda.

```
aws iotanalytics create-dataset-content --dataset-name my_dataset
```

Isi set data

Hasil query SQL adalah konten dataset Anda, disimpan sebagai file, dalam format CSV. File ini tersedia untuk Anda melalui Amazon S3. Langkah-langkah berikut menunjukkan bagaimana Anda dapat memeriksa apakah hasil Anda sudah siap dan mengunduh file.

Topik

- [Mengakses konten dataset di AWS IoT Analytics \(Konsol\)](#)
- [Mengakses konten dataset di AWS IoT Analytics \(AWS CLI\)](#)

Mengakses konten dataset di AWS IoT Analytics (Konsol)

Jika dataset Anda berisi data apa pun, Anda dapat melihat pratinjau dan mengunduh hasil kueri SQL Anda AWS IoT Analytics konsol.

Untuk mengakses AWS IoT Analytics hasil set data

1. Di konsol, di **Set Data** halaman, pilih nama set data yang ingin Anda akses.
2. Pada halaman ringkasan dataset, pilih **Daftar isi** tab.
3. Di **Isi set data** tabel, pilih nama query yang ingin Anda pratinjau hasil atau download file csv hasil.

Mengakses konten dataset di AWS IoT Analytics (AWS CLI)

Jika dataset Anda berisi data apa pun, Anda dapat melihat pratinjau dan mengunduh hasil kueri SQL Anda.

Contoh yang ditunjukkan di sini menggunakan AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya tentang AWS CLI, lihat [Panduan Pengguna AWS Command Line](#)

[Interface](#). Untuk informasi selengkapnya tentang perintah CLI yang tersedia AWS IoT Analytics, lihat [iotanalytics](#) di AWS Command Line Interface Referensi.

Untuk mengakses AWS IoT Analytics hasil set data (AWS CLI)

1. Jalankan berikut `get-dataset-content` perintah untuk melihat hasil kueri Anda.

```
aws iotanalytics get-dataset-content --dataset-name my_iotsitewise_dataset
```

2. Jika dataset Anda berisi data apapun, maka output dari `get-dataset-content`, memiliki `"state": "SUCCEEDED"` distatus bidang, seperti pada contoh berikut ini.

```
{
  "timestamp": 1508189965.746,
  "entries": [
    {
      "entryName": "my_entry_name",
      "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-
Amz-"
    }
  ],
  "status": {
    "state": "SUCCEEDED",
    "reason": "A useful comment."
  }
}
```

3. Output dari `get-dataset-content` termasuk `dataURI`, yang merupakan URL ditandatangani untuk hasil output. Hal ini berlaku untuk waktu yang singkat (beberapa jam). Kunjungi `dataURI` URL untuk mengakses hasil kueri SQL Anda.

Note

Tergantung pada alur kerja Anda, Anda mungkin ingin selalu menelepon `get-dataset-content` sebelum Anda mengakses konten karena memanggil perintah ini menghasilkan URL baru yang ditandatangani.

Tutorial: KueriAWS IoT SiteWisedata diAWS IoT Analytics

Tutorial ini menunjukkan bagaimana kueriAWS IoT SiteWisedata diAWS IoT Analytics. Tutorial menggunakan data dari demo diAWS IoT SiteWiseyang menyediakan satu set sampel data untuk peternakan angin.

Important

Anda akan dikenakan biaya untuk sumber daya yang dibuat dan dikonsumsi demo ini.

Topik

- [Prasyarat](#)
- [Memuat dan memverifikasi data](#)
- [Eksplorasi data](#)
- [Jalankan kueri statistik](#)
- [Membersihkan sumber daya tutorial Anda](#)

Prasyarat

Dalam tutorial ini, Anda akan memerlukan sumber daya berikut:

- Anda harus memilikiAWS Akun untuk memulaiAWS IoT SiteWisedanAWS IoT Analytics. Jika Anda sama sekali belum memilikinya, ikuti prosedurnya[Untuk membuatAWS Akun](#).
- Komputer pengembangan yang menjalankan Windows, macOS, Linux, atau Unix untuk mengaksesAWS Management Console. Untuk informasi lebih lanjut, lihat [Memulai dengan AWS Management Console](#).
- AWS IoT SiteWisedata yang mendefinisikanAWS IoT SiteWisemodel dan aset dan aliran data yang mewakili data dari peralatan pertanian angin. Untuk membuat data Anda, ikuti langkah-langkah di[MembuatAWS IoT SiteWisedemo](#)di dalamAWS IoT SiteWisePanduan Pengguna.
- KlasterAWS IoT SiteWisedemo angin data peralatan pertanian di toko data yang ada yang Anda kelola. Untuk informasi lebih lanjut tentang cara membuat penyimpanan data untuk AndaAWS IoT SiteWisedata, lihat[Konfigurasi pengaturan penyimpanan](#)di dalamAWS IoT SiteWisePanduan Pengguna.

Note

Klaster AWS IoT SiteWise metadata muncul di AWS IoT SiteWise penyimpanan data segera setelah pembuatan; namun, dibutuhkan waktu hingga enam jam agar data mentah Anda muncul. Pada saat yang sama, Anda dapat membuat AWS IoT Analytics dataset dan menjalankan kueri pada metadata Anda.

Langkah selanjutnya

Memuat dan memverifikasi data

Memuat dan memverifikasi data

Data yang Anda kueri dalam tutorial ini adalah kumpulan sampel AWS IoT SiteWise data yang memodelkan turbin mesin angin di ladang angin.

Note

Anda akan query tiga tabel di toko data Anda di seluruh tutorial ini:

- `raw`- Berisi data mentah dan belum diproses untuk setiap aset.
- `asset_metadata`- Berisi informasi umum tentang setiap aset.
- `asset_hierarchy_metadata`- Berisi informasi tentang hubungan antara aset.

Untuk menjalankan query SQL dalam tutorial ini

1. Ikuti langkah-langkah di [Membuat set data AWS IoT SiteWise data \(Konsol\)](#) atau [Membuat set data AWS IoT SiteWise data AWS CLI](#) untuk membuat AWS IoT Analytics set data Anda AWS IoT SiteWise data.
2. Untuk memperbarui kueri dataset Anda di seluruh tutorial ini, lakukan hal berikut.
 - a. Di AWS IoT Analytics konsol, pada Set Data halaman, memilih nama set data yang Anda buat pada halaman sebelumnya.
 - b. Pada halaman ringkasan set data, pilih edit untuk mengedit kueri SQL Anda.
 - c. Untuk menampilkan hasil dalam tabel mengikuti kueri, pilih Kueri pengujian.

Atau, Anda dapat menjalankan hal berikut update-dataset perintah untuk memodifikasi query SQL dengan AWS CLI.

```
aws iotanalytics update-dataset --cli-input-json file://update-query.json
```

Isi dari update-query.json:

```
{
  "datasetName": "my_dataset",
  "actions": [
    {
      "actionName": "myDatasetUpdateAction",
      "queryAction": {
        "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
LIMIT 3"
      }
    }
  ]
}
```

- Di AWS IoT Analytics konsol atau dengan AWS CLI, jalankan kueri berikut pada data Anda untuk memverifikasi bahwa Anda asset_metadata tabel dimuat berhasil.

```
SELECT COUNT(*) FROM my_iotsitewise_datastore.asset_metadata
```

Dengan cara yang sama, Anda dapat memverifikasi bahwa asset_hierarchy_metadata dan rawTable tidak kosong.

Langkah Selanjutnya

[Eksplorasi data](#)

Eksplorasi data

Setelah Anda AWS IoT SiteWise data dibuat dan dimuat ke dalam penyimpanan data, Anda dapat membuat AWS IoT Analytics dataset dan menjalankan kueri SQL di AWS IoT Analytics untuk menemukan wawasan tentang aset Anda. Kueri berikut menunjukkan bagaimana Anda dapat menjelajahi data Anda sebelum menjalankan kueri statistik.

Untuk menjelajahi data Anda dengan kueri SQL

1. Lihat contoh kolom dan nilai di setiap tabel, seperti di tabel mentah.

```
SELECT * FROM my_iotsitewise_datastore.raw LIMIT 5
```

2. Gunakan `SELECT DISTINCT` untuk query aset_metadata tabel dan daftar (unik) nama-nama Anda AWS IoT SiteWise aset.

```
SELECT DISTINCT assetname FROM my_iotsitewise_datastore.asset_metadata ORDER BY
assetname
```

3. Untuk mencantumkan informasi tentang properti untuk tertentu AWS IoT SiteWise aset, gunakan `WHERE` klausa.

```
SELECT assetpropertyname,
       assetpropertyunit,
       assetpropertydatatype
FROM my_iotsitewise_datastore.asset_metadata
WHERE assetname = 'Demo Turbine Asset 2'
```

4. Dengan AWS IoT Analytics, Anda dapat menggabungkan data dari dua atau lebih tabel di penyimpanan data Anda, seperti pada contoh berikut.

```
SELECT * FROM my_iotsitewise_datastore.raw AS raw
JOIN my_iotsitewise_datastore.asset_metadata AS asset_metadata
ON raw.seriesId = asset_metadata.timeseriesId
```

Untuk melihat semua hubungan antara aset Anda, gunakan `JOIN` fungsi dalam kueri berikut.

```
SELECT DISTINCT parent.assetName as "Parent name",
       child.assetName AS "Child name"
FROM (
  SELECT sourceAssetId AS parent,
         targetAssetId AS child
  FROM my_iotsitewise_datastore.asset_hierarchy_metadata
  WHERE associationType = 'CHILD'
)
AS relations
JOIN my_iotsitewise_datastore.asset_metadata AS child
ON relations.child = child.assetId
```

```
JOIN my_iotsitewise_datastore.asset_metadata AS parent
ON relations.parent = parent.assetId
```

Langkah selanjutnya

[Jalankan kueri statistik](#)

Jalankan kueri statistik

Sekarang bahwa Anda telah menjelajahi AndaAWS IoT SiteWisedata, Anda dapat menjalankan kueri statistik yang memberikan wawasan berharga untuk peralatan industri Anda. Kueri berikut menunjukkan beberapa informasi yang dapat Anda ambil.

Untuk menjalankan kueri statistikAWS IoT SiteWiseData Peternakan Angin Demo

1. Jalankan perintah SQL berikut untuk menemukan nilai terbaru dari semua properti dengan nilai numerik untuk aset tertentu (Demo Turbine Asset 4).

```
SELECT assetName,
       assetPropertyName,
       assetPropertyUnit,
       max_by(value, timeInSeconds) AS Latest
FROM (
  SELECT *,
         CASE assetPropertyDataType
           WHEN 'DOUBLE' THEN
             cast(doubleValue AS varchar)
           WHEN 'INTEGER' THEN
             cast(integerValue AS varchar)
           WHEN 'STRING' THEN
             stringValue
           WHEN 'BOOLEAN' THEN
             cast(booleanValue AS varchar)
           ELSE NULL
         END AS value
  FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
  JOIN my_iotsitewise_datastore.raw AS raw
    ON raw.seriesId = asset_metadata.timeSeriesId
  WHERE startYear=2021
         AND startMonth=7
         AND startDay=8
         AND assetName='Demo Turbine Asset 4'
```

```
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit
```

2. Bergabunglah dengan tabel metadata dan tabel mentah Anda untuk mengidentifikasi properti kecepatan angin maksimum untuk semua aset, selain aset induknya.

```
SELECT child_assets_data_set.parentAssetId,
       child_assets_data_set.childAssetId,
       asset_metadata.assetPropertyId,
       asset_metadata.assetPropertyName,
       asset_metadata.timeSeriesId,
       raw_data_set.max_speed
FROM (
  SELECT sourceAssetId AS parentAssetId,
         targetAssetId AS childAssetId
  FROM my_iotsitewise_datastore.asset_hierarchy_metadata
  WHERE associationType = 'CHILD'
)
AS child_assets_data_set
JOIN mls_demo.asset_metadata AS asset_metadata
  ON asset_metadata.assetId = child_assets_data_set.childAssetId
JOIN (
  SELECT seriesId, MAX(doubleValue) AS max_speed
  FROM my_iotsitewise_datastore.raw
  GROUP BY seriesId
)
AS raw_data_set
ON raw_data_set.seriesId = asset_metadata.timeseriesid
WHERE assetPropertyName = 'Wind Speed'
ORDER BY max_speed DESC
```

3. Untuk menemukan nilai rata-rata properti tertentu (Kecepatan Angin) untuk aset (Demo Turbine Asset 2), jalankan perintah SQL berikut. Anda harus menggantinya dengan ID ember Anda.

```
SELECT AVG(doubleValue) as "Average wind speed"
FROM my_iotsitewise_datastore.raw
WHERE seriesId =
  (SELECT timeseriesId
   FROM my_iotsitewise_datastore.asset_metadata as asset_metadata
   WHERE asset_metadata.assetname = 'Demo Turbine Asset 2'
        AND asset_metadata.assetpropertyname = 'Wind Speed')
```


Langkah selanjutnya

[Membersihkan sumber daya tutorial Anda](#)

Membersihkan sumber daya tutorial Anda

Setelah Anda menyelesaikan tutorial, bersihkan sumber daya Anda untuk menghindari biaya.

Untuk menghapus AWS IoT SiteWise demo

Klaster AWS IoT SiteWise demo menghapus dirinya sendiri setelah seminggu. Jika Anda selesai menggunakan sumber daya demo, Anda dapat menghapus demo sebelumnya. Untuk menghapus demo secara manual, gunakan langkah-langkah berikut.

1. Navigasikan ke [konsol AWS CloudFormation](#) tersebut.
2. Pilih `IoTSiteWiseDemoAssets` dari daftar Tumpukan.
3. Pilih Delete (Hapus). Ketika Anda menghapus tumpukan, semua sumber daya yang dibuat untuk demo dihapus.
4. Pada dialog konfirmasi, masukkan Hapus.

Tumpukan membutuhkan waktu sekitar 15 menit untuk menghapus. 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.

Untuk menghapus penyimpanan data Anda


- Untuk menghapus penyimpanan data terkelola Anda, jalankan perintah `aws iotanalytics delete-datastore`, seperti pada contoh berikut.

```
aws iotanalytics delete-datastore --datastore-name my_IotSiteWise_datastore
```

Untuk menghapus AWS IoT Analytics dataset

- Untuk menghapus dataset Anda, jalankan perintah `aws iotanalytics delete-dataset`, seperti pada contoh berikut. Anda tidak perlu menghapus konten set data sebelum Anda melakukan operasi ini.

```
aws iotanalytics delete-dataset --dataset-name my_dataset
```

 Note

Perintah ini tidak menghasilkan output.

Aktivitas ALUr

Pipa fungsional yang paling sederhana menghubungkan saluran ke penyimpanan data, yang menjadikannya pipa dengan dua aktivitas: `channel` aktivitas dan `datastore` aktivitas. Anda dapat mencapai pemrosesan pesan yang lebih kuat dengan menambahkan aktivitas tambahan ke pipeline Anda.

Anda dapat menggunakan [RunPipelineActivity](#) operasi untuk mensimulasikan hasil menjalankan aktivitas pipeline pada payload pesan yang Anda berikan. Anda mungkin menemukan ini berguna ketika Anda mengembangkan dan men-debug aktivitas pipeline Anda. [RunPipelineActivity contoh](#) menunjukkan bagaimana itu digunakan.

Aktivitas saluran

Kegiatan pertama dalam pipa harus `channel` aktivitas yang menentukan sumber pesan yang akan diproses.

```
{
  "channel": {
    "name": "MyChannelActivity",
    "channelName": "mychannel",
    "next": "MyLambdaActivity"
  }
}
```

Aktivitas Datastore

Kluster `datastore` aktivitas, yang menentukan di mana harus menyimpan data yang diproses, adalah aktivitas terakhir.

```
{
  "datastore": {
    "name": "MyDatastoreActivity",
    "datastoreName": "mydatastore"
  }
}
```

AWS Lambdaaktivitas

Anda dapat menggunakan **Lambda**aktivitas untuk melakukan pemrosesan yang kompleks pada pesan. Misalnya, Anda dapat memperkaya pesan dengan data dari output operasi API eksternal, atau memfilter pesan berdasarkan logika dari Amazon DynamoDB. Namun, Anda tidak dapat menggunakan aktivitas pipeline ini untuk menambahkan pesan tambahan, atau menghapus pesan yang ada, sebelum memasuki penyimpanan data.

Klaster AWS Lambda fungsi yang digunakan dalam **Lambda**aktivitas harus menerima dan mengembalikan array objek JSON. Sebagai contoh, lihat [the section called “Contoh fungsi Lambda 1”](#).

Untuk memberikan izin AWS IoT Analytics izin untuk mengaktifkan fungsi Lambda Anda, Anda harus menambahkan kebijakan. Misalnya, gunakan perintah CLI berikut dan ganti *exampleFunctionName* dengan nama fungsi Lambda Anda, ganti *123456789012* dengan AWSID Akun, dan gunakan Nama Sumber Daya Amazon (ARN) dari pipeline yang mengaktifkan fungsi Lambda yang diberikan.

```
aws lambda add-permission --function-name exampleFunctionName --  
action lambda:InvokeFunction --statement-id iotanalytics --principal  
iotanalytics.amazonaws.com --source-account 123456789012 --source-arn  
arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline
```

Perintah mengembalikan hal berikut:

```
{  
  "Statement": [{"Sid": "iotanalytica", "Effect": "Allow",  
    "Principal": {"Service": "iotanalytics.amazonaws.com"}, "Action":  
    "lambda:InvokeFunction", "Resource": "arn:aws:lambda:aws-region:aws-  
account:function:exampleFunctionName", "Condition": {"StringEquals":  
    {"AWS:SourceAccount": "123456789012"}, "ArnLike": {"AWS:SourceArn":  
    "arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline"}}}]  
}
```

Untuk informasi selengkapnya, lihat [Menggunakan kebijakan berbasis sumber daya untuk AWS Lambda](#) di dalam AWS Lambda Panduan Pengembang.

Contoh fungsi Lambda 1

Dalam contoh ini, fungsi Lambda menambahkan informasi berdasarkan data dalam pesan asli. Perangkat memublikasikan pesan dengan payload yang mirip dengan contoh berikut.

```
{
  "thingid": "00001234abcd",
  "temperature": 26,
  "humidity": 29,
  "location": {
    "lat": 52.4332935,
    "lon": 13.231694
  },
  "ip": "192.168.178.54",
  "datetime": "2018-02-15T07:06:01"
}
```

Dan perangkat memiliki definisi pipa berikut.

```
{
  "pipeline": {
    "activities": [
      {
        "channel": {
          "channelName": "foobar_channel",
          "name": "foobar_channel_activity",
          "next": "lambda_foobar_activity"
        }
      },
      {
        "lambda": {
          "lambdaName": "MyAnalyticsLambdaFunction",
          "batchSize": 5,
          "name": "lambda_foobar_activity",
          "next": "foobar_store_activity"
        }
      }
    ],
    "datastore": {
      "datastoreName": "foobar_datastore",
      "name": "foobar_store_activity"
    }
  }
}
```

```
    ],
    "name": "foobar_pipeline",
    "arn": "arn:aws:iotanalytics:eu-west-1:123456789012:pipeline/foobar_pipeline"
  }
}
```

Berikut Lambda Python fungsi (MyAnalyticsLambdaFunction) menambahkan URL GMaps dan suhu, di Fahrenheit, ke pesan.

```
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def c_to_f(c):
    return 9.0/5.0 * c + 32

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))
    maps_url = 'N/A'

    for e in event:
        #e['foo'] = 'addedByLambda'
        if 'location' in e:
            lat = e['location']['lat']
            lon = e['location']['lon']
            maps_url = "http://maps.google.com/maps?q={},{}".format(lat,lon)

        if 'temperature' in e:
            e['temperature_f'] = c_to_f(e['temperature'])

        logger.info("maps_url: {}".format(maps_url))
        e['maps_url'] = maps_url

    logger.info("event after processing: {}".format(event))

    return event
```

Contoh fungsi Lambda 2

Teknik yang berguna adalah mengompres dan membuat serial muatan pesan untuk mengurangi biaya transportasi dan penyimpanan. Dalam contoh kedua ini, fungsi Lambda mengasumsikan bahwa payload pesan mewakili asli JSON, yang telah dikompresi dan kemudian dikodekan base64 (serial) sebagai string. Ia mengembalikan JSON asli.

```
import base64
import gzip
import json
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def decode_to_bytes(e):
    return base64.b64decode(e)

def decompress_to_string(binary_data):
    return gzip.decompress(binary_data).decode('utf-8')

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))

    decompressed_data = []

    for e in event:
        binary_data = decode_to_bytes(e)
        decompressed_string = decompress_to_string(binary_data)

        decompressed_data.append(json.loads(decompressed_string))

    logger.info("event after processing: {}".format(decompressed_data))

    return decompressed_data
```

AddAttributes aktivitas

Sesi `addAttributes` aktivitas menambahkan atribut berdasarkan atribut yang ada dalam pesan. Ini memungkinkan Anda mengubah bentuk pesan sebelum disimpan. Misalnya, Anda dapat menggunakan `addAttributes` untuk menormalkan data yang berasal dari berbagai generasi firmware perangkat.

Pertimbangkan pesan input berikut.

```
{
  "device": {
    "id": "device-123",
    "coord": [ 47.6152543, -122.3354883 ]
  }
}
```

Kluster `addAttributes` aktivitas terlihat seperti berikut.

```
{
  "addAttributes": {
    "name": "MyAddAttributesActivity",
    "attributes": {
      "device.id": "id",
      "device.coord[0]": "lat",
      "device.coord[1]": "lon"
    },
    "next": "MyRemoveAttributesActivity"
  }
}
```

Aktivitas ini memindahkan ID perangkat ke tingkat root dan mengekstrak nilai `coord` array, mempromosikan mereka ke atribut tingkat atas yang disebut `lat` dan `lon`. Sebagai hasil dari aktivitas ini, pesan masukan ditransformasikan ke contoh berikut.

```
{
  "device": {
    "id": "device-123",
    "coord": [ 47.6, -122.3 ]
  },
  "id": "device-123",
```



```
"lat": 47.6,  
"lon": -122.3  
}
```

Atribut perangkat asli masih ada. Jika Anda ingin menghapusnya, Anda dapat menggunakan `removeAttributes` aktivitas.

RemoveAttributes aktivitas

SEBUAH `removeAttributes` aktivitas menghapus atribut dari pesan. Misalnya, diberikan pesan yang merupakan hasil dari `addAttributes` aktivitas.

```
{  
  "device": {  
    "id": "device-123",  
    "coord": [ 47.6, -122.3 ]  
  },  
  "id": "device-123",  
  "lat": 47.6,  
  "lon": -122.3  
}
```

Untuk menormalkan pesan itu sehingga hanya mencakup data yang diperlukan pada tingkat root, gunakan yang berikut `removeAttributes` aktivitas.

```
{  
  "removeAttributes": {  
    "name": "MyRemoveAttributesActivity",  
    "attributes": [  
      "device"  
    ],  
    "next": "MyDatastoreActivity"  
  }  
}
```

Hal ini menyebabkan pesan berikut yang mengalir di sepanjang pipa.

```
{  
  "id": "device-123",  
  "lat": 47.6,  
}
```

```
"lon": -122.3  
}
```

SelectAttributes aktivitas

KlasterselectAttributesAktivitas membuat pesan baru hanya menggunakan atribut yang ditentukan dari pesan asli. Setiap atribut lainnya dijatuhkan.selectAttributesmenciptakan atribut baru di bawah akar pesan saja. Jadi mengingat pesan ini:

```
{  
  "device": {  
    "id": "device-123",  
    "coord": [ 47.6152543, -122.3354883 ],  
    "temp": 50,  
    "hum": 40  
  },  
  "light": 90  
}
```

dan kegiatan ini:

```
{  
  "selectAttributes": {  
    "name": "MySelectAttributesActivity",  
    "attributes": [  
      "device.temp",  
      "device.hum",  
      "light"  
    ],  
    "next": "MyDatastoreActivity"  
  }  
}
```

Hasilnya adalah pesan berikut yang mengalir melalui pipa.

```
{  
  "temp": 50,  
  "hum": 40,  
  "light": 90  
}
```

Sekali lagi, `selectAttributes` hanya dapat membuat objek tingkat akar.

Memfilter aktivitas

SEBUAH `filter` aktivitas memfilter pesan berdasarkan atributnya. Ekspresi yang digunakan dalam aktivitas ini terlihat seperti `SQLWHERE` klausa, yang harus mengembalikan Boolean.

```
{
  "filter": {
    "name": "MyFilterActivity",
    "filter": "temp > 40 AND hum < 20",
    "next": "MyDatastoreActivity"
  }
}
```

DeviceRegistryEnrich aktivitas

Klaster `deviceRegistryEnrich` aktivitas memungkinkan Anda untuk menambahkan data dari `AWS IoT Registry` perangkat ke payload pesan Anda. Misalnya, diberikan pesan berikut:

```
{
  "temp": 50,
  "hum": 40,
  "device" {
    "thingName": "my-thing"
  }
}
```

dand `deviceRegistryEnrich` aktivitas yang terlihat seperti ini:

```
{
  "deviceRegistryEnrich": {
    "name": "MyDeviceRegistryEnrichActivity",
    "attribute": "metadata",
    "thingName": "device.thingName",
    "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
    "next": "MyDatastoreActivity"
  }
}
```

Pesan output sekarang terlihat seperti contoh ini.

```
{
  "temp" : 50,
  "hum" : 40,
  "device" {
    "thingName" : "my-thing"
  },
  "metadata" : {
    "defaultClientId": "my-thing",
    "thingTypeName": "my-thing",
    "thingArn": "arn:aws:iot:us-east-1:<your-account-number>:thing/my-thing",
    "version": 1,
    "thingName": "my-thing",
    "attributes": {},
    "thingId": "aaabbbccc-dddeef-gghh-jjkk-llmmnnoopp"
  }
}
```

Anda harus menentukan peran `roleArn` bidang definisi aktivitas yang memiliki izin yang sesuai terlampir. Peran harus memiliki kebijakan izin yang terlihat seperti contoh berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeThing"
      ],
      "Resource": [
        "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
      ]
    }
  ]
}
```

dan kebijakan kepercayaan yang terlihat seperti:

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

```

    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}

```

DeviceShadowEnrich aktivitas

SEBUAH `deviceShadowEnrich` aktivitas menambahkan informasi dari AWS IoT Layanan Device Shadow ke pesan. Misalnya, diberikan pesan:

```

{
  "temp": 50,
  "hum": 40,
  "device": { "thingName": "my-thing" }
}

```

dan yang berikut `deviceShadowEnrich` aktifitas aktivitas

```

{
  "deviceShadowEnrich": {
    "name": "MyDeviceShadowEnrichActivity",
    "attribute": "shadow",
    "thingName": "device.thingName",
    "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
    "next": "MyDatastoreActivity"
  }
}

```

Hasilnya adalah pesan yang terlihat seperti berikut contoh berikut.

```

{
  "temp": 50,
  "hum": 40,

```

```

"device": {
  "thingName": "my-thing"
},
"shadow": {
  "state": {
    "desired": {
      "attributeX": valueX, ...
    },
    "reported": {
      "attributeX": valueX, ...
    },
    "delta": {
      "attributeX": valueX, ...
    }
  },
  "metadata": {
    "desired": {
      "attribute1": {
        "timestamp": timestamp
      }, ...
    },
    "reported": ": {
      "attribute1": {
        "timestamp": timestamp
      }, ...
    }
  },
  "timestamp": timestamp,
  "clientToken": "token",
  "version": version
}
}

```

Anda harus menentukan peran `roleArn` bidang definisi aktivitas yang memiliki izin yang sesuai terlampir. Peran harus memiliki kebijakan izin yang terlihat seperti berikut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:GetThingShadow"

```

```

    ],
    "Resource": [
      "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
    ]
  }
]
}

```

dan kebijakan kepercayaan yang terlihat seperti:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}

```

Aktivitas Matematika

SEBUAHmathaktivitas mengomputasi ekspresi aritmetika menggunakan atribut pesan. Ekspresi harus mengembalikan angka. Misalnya, diberikan pesan input berikut:

```

{
  "tempF": 50,
}

```

setelah diproses oleh berikutmathaktifitas aktivitas

```

{
  "math": {
    "name": "MyMathActivity",
    "math": "(tempF - 32) / 2",
  }
}

```

```
    "attribute": "tempC",
    "next": "MyDatastoreActivity"
  }
}
```

pesan yang dihasilkan terlihat seperti:

```
{
  "tempF" : 50,
  "tempC": 9
}
```

Operator aktivitas matematika dan fungsi

Anda dapat menggunakan operator berikut di dalam aktivitas

+	tambahan
-	Pengurangan
*	perkalian
/	Pembagian
%	modulo

Anda dapat menggunakan fungsi berikut di dalam aktivitas

- [abs \(Desimal\)](#)
- [acos \(Desimal\)](#)
- [asin \(Desimal\)](#)
- [atan \(Desimal\)](#)
- [atan2 \(Desimal, Desimal\)](#)
- [ceil \(Desimal\)](#)
- [cos \(Desimal\)](#)

- [cosh \(Desimal\)](#)
- [exp \(Desimal\)](#)
- [ln \(Desimal\)](#)
- [log \(Desimal\)](#)
- [mod \(Desimal, Desimal\)](#)
- [kekuasaan \(Desimal, Desimal\)](#)
- [bulat \(Desimal\)](#)
- [tanda \(Desimal\)](#)
- [sin \(Desimal\)](#)
- [sinh \(Desimal\)](#)
- [sqrt \(Desimal\)](#)
- [tan \(Desimal\)](#)
- [tanh \(Desimal\)](#)
- [trunc \(Desimal, Integer\)](#)

abs (Desimal)

Mengembalikan nilai absolut dari angka.

Contoh: `abs (-5)` mengembalikan 5

Tipe Argumen	Hasil
Int	Int, nilai absolut dari argumen.
Decimal	Decimal, nilai absolut dari argumen
Boolean	Undefined .
String	Decimal. Hasilnya adalah nilai absolut dari argumen. Jika string tidak dapat dikonversi, hasilnya adalah Undefined .
Susunan	Undefined .

Tipe Argumen	Hasil
Objek	Undefined .
Nol	Undefined .
Tidak terdefinisi	Undefined .

acos (Desimal)

Mengembalikan cosinus terbalik dari angka dalam radian. DecimaI argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: $\text{acos}(0) = 1.5707963267948966$

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), kosinus terbalik dari argumen. Hasil imajiner dikembalikan sebagaiUndefined .
Decimal	Decimal(dengan presisi ganda), kosinus terbalik dari argumen. Hasil imajiner dikembalikan sebagaiUndefined .
Boolean	Undefined .
String	Decimal(dengan presisi ganda) kosinus terbalik dari argumen. Jika string tidak dapat dikonversi, hasilnya adalahUndefined . Hasil imajiner dikembalikan sebagaiUndefined .
Susunan	Undefined .
Objek	Undefined .
Nol	Undefined .
Tidak terdefinisi	Undefined .

asin (Desimal)

Mengembalikan sinus terbalik dari angka dalam radian. `Decimal` argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `asin(0) = 0,0`

Tipe Argumen	Hasil
Int	<code>Decimal</code> (dengan presisi ganda), sinus terbalik dari argumen. Hasil imajiner dikembalikan sebagai <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (dengan presisi ganda), sinus terbalik dari argumen. Hasil imajiner dikembalikan sebagai <code>Undefined</code> .
Boolean	<code>Undefined</code> .
String	<code>Decimal</code> (dengan presisi ganda), sinus terbalik dari argumen. Jika string tidak dapat dikonversi, hasilnya adalah <code>Undefined</code> . Hasil imajiner dikembalikan sebagai <code>Undefined</code> .
Susunan	<code>Undefined</code> .
Objek	<code>Undefined</code> .
Nol	<code>Undefined</code> .
Tidak terdefinisi	<code>Undefined</code> .

atan (Desimal)

Mengembalikan garis singgung terbalik dari angka dalam radian. `Decimal` argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `atan(0) = 0,0`

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), garis singgung terbalik dari argumen. Hasil imajiner dikembalikan sebagaiUndefined .
Decimal	Decimal(dengan presisi ganda), garis singgung terbalik dari argumen. Hasil imajiner dikembalikan sebagaiUndefined .
Boolean	Undefined .
String	Decimal(dengan presisi ganda), garis singgung terbalik dari argumen. Jika string tidak dapat dikonversi, hasilnya adalahUndefined . Hasil imajiner dikembalikan sebagaiUndefined .
Susunan	Undefined .
Objek	Undefined .
Nol	Undefined .
Tidak terdefinisi	Undefined .

atan2 (Desimal, Desimal)

Mengembalikan sudut, dalam radian, antara sumbu x positif dan titik (x, y) didefinisikan dalam dua argumen. Sudut positif untuk sudut berlawanan arah jarum jam (setengah bidang atas, $y > 0$), dan negatif untuk sudut searah jarum jam. Hasil argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: $\text{atan}(1, 0) = 1.5707963267948966$

Tipe Argumen	Tipe Argumen	Hasil
Int / Decimal	Int / Decimal	Decimal(dengan presisi ganda), sudut antara sumbu x dan titik yang ditentukan (x, y)
Int / Decimal / String	Int / Decimal / String	Decimal, garis singgung terbalik dari titik yang dijelaskan. Jika string tidak dapat dikonversi, hasilnya adalah <code>Undefined</code> .
Nilai Lainnya	Nilai Lainnya	<code>Undefined</code> .

ceil (Desimal)

Membulatkan yang diberikan `Decimal` hingga ke yang terdekat `Int`.

Contoh:

`ceil(1.2)= 2`

`ceil(11.2)= -1`

Tipe Argumen	Hasil
Int	Int, nilai argumen.
Decimal	Int, string dikonversi ke <code>Decimal</code> dan dibulatkan ke yang terdekat <code>Int</code> . Jika string tidak dapat dikonversi ke <code>Decimal</code> , hasilnya adalah <code>Undefined</code> .
Nilai Lainnya	<code>Undefined</code> .

cos (Desimal)

Mengembalikan kosinus angka dalam radian. `Decimal` argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `cos(0) = 1`

Tipe Argumen	Hasil
Int	<code>Decimal</code> (dengan presisi ganda), kosinus argumen. Hasil imajiner dikembalikan sebagai <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (dengan presisi ganda), kosinus argumen. Hasil imajiner dikembalikan sebagai <code>Undefined</code> .
Boolean	<code>Undefined</code> .
String	<code>Decimal</code> (dengan presisi ganda), kosinus argumen. Jika string tidak dapat dikonversi ke <code>Decimal</code> , hasilnya adalah <code>Undefined</code> . Hasil imajiner dikembalikan sebagai <code>Undefined</code> .
Susunan	<code>Undefined</code> .
Objek	<code>Undefined</code> .
Nol	<code>Undefined</code> .
Tidak terdefinisi	<code>Undefined</code> .

cosh (Desimal)

Mengembalikan kosinus hiperbolik angka dalam radian. `Decimal` argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `cosh(2.3) = 5.037220649268761`

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), kosinus hiperbolik argumen. Hasil imajiner dikembalikan sebagaiUndefined .
Decimal	Decimal(dengan presisi ganda), kosinus hiperbolik argumen. Hasil imajiner dikembalikan sebagaiUndefined .
Boolean	Undefined .
String	Decimal(dengan presisi ganda), kosinus hiperbolik argumen. Jika string tidak dapat dikonversi keDecimal, hasilnya adalahUndefined . Hasil imajiner dikembalikan sebagaiUndefined .
Susunan	Undefined .
Objek	Undefined .
Nol	Undefined .
Tidak terdefinisi	Undefined .

exp (Desimal)

Pengembalian diangkat ke argumen desimal.Decimal argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: $\exp(1) = 1$

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), e^{\wedge} argumen.
Decimal	Decimal(dengan presisi ganda), e^{\wedge} argumen

Tipe Argumen	Hasil
String	Decimal(dengan presisi ganda), e^{argumen} . Jika String tidak dapat dikonversi ke Decimal, hasilnya jika Undefined .
Nilai Lainnya	Undefined .

In (Desimal)

Mengembalikan logaritma natural argumen. Decimal argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: $\ln(e) = 1$

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), log alami argumen.
Decimal	Decimal(dengan presisi ganda), log alami argumen
Boolean	Undefined .
String	Decimal(dengan presisi ganda), log alami argumen. Jika string tidak dapat dikonversi ke Decimal hasilnya Undefined .
Susunan	Undefined .
Objek	Undefined .
Nol	Undefined .
Tidak terdefinisi	Undefined .

log (Desimal)

Mengembalikan logaritma basis 10 dari argumen. `Decimal` argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `log(100) = 2.0`

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), dasar 10 log argumen.
Decimal	Decimal(dengan presisi ganda), dasar 10 log argumen.
Boolean	Undefined .
String	Decimal(dengan presisi ganda), dasar 10 log argumen. Jika String tidak dapat dikonversi ke Decimal, hasilnya adalah Undefined .
Susunan	Undefined .
Objek	Undefined .
Nol	Undefined .
Tidak terdefinisi	Undefined .

mod (Desimal, Desimal)

Mengembalikan sisa pembagian argumen pertama dari argumen kedua. Anda juga dapat menggunakan `%` sebagai operator infix untuk fungsi modulo yang sama.

Contoh: `mod(8, 3) = 3`

Operan Kiri	Operan kanan	Output
Int	Int	Int, modulo argumen pertama dari argumen kedua.
Int / Decimal	Int / Decimal	Decimal, modulo argumen pertama dari argumen kedua.
String / Int / Decimal	String / Int / Decimal	Jika semua string dikonversi keDecimals, Hasil jika argumen pertama memodulasi argumen kedua. Jika tidak, Undefined .
Nilai Lainnya	Nilai Lainnya	Undefined .

kekuasaan (Desimal, Desimal)

Mengembalikan argumen pertama diangkat ke argumen kedua.Decimalargumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `power(2, 5) = 32.0`

Tipe Argumen 1	Tipe Argumen 2	Output
Int / Decimal	Int / Decimal	SEBUAHDecimal(dengan presisi ganda), argumen pertama diangkat ke kekuatan argumen kedua.
Int / Decimal / String	Int / Decimal / String	SEBUAHDecimal(dengan presisi ganda), argumen pertama diangkat ke kekuatan argumen kedua. Setiap string dikonversi keDecimals. Jika ada caraStringgagal

Tipe Argumen 1	Tipe Argumen 2	Output
		untuk dikonversi keDecimal, hasilnya adalahUndefined .
Nilai Lainnya	Nilai Lainnya	Undefined .

bulat (Desimal)

Membulatkan yang diberikanDecimalke yang terdekatInt. JikaDecimalberjarak sama dari duaIntnilai (misalnya, 0,5),Decimaldibulatkan ke atas.

Contoh:

Round(1.2)= 1

Round(1.5)= 2

Round(1.7)= 2

Round(-1.1)= -1

Round(-1.5)= -2

Tipe Argumen	Hasil
Int	Argumen
Decimal	Decimaldibulatkan ke arah terdekatInt.
String	Decimaldibulatkan ke arah terdekatInt. Jika string tidak dapat dikonversi keDecimal, hasilnya adalahUndefined .
Nilai Lainnya	Undefined .

tanda (Desimal)

Mengembalikan tanda nomor yang diberikan. Ketika tanda argumen positif, 1 dikembalikan. Ketika tanda argumen negatif, -1 dikembalikan. Jika argumen adalah 0, 0 dikembalikan.

Contoh:

$\text{sign}(-7) = -1$

$\text{sign}(0) = 0 = 0$

$\text{sign}(13) = 1$

Tipe Argumen	Hasil
Int	Int, tandaIntnilai.
Decimal	Int, tandaDecimalnilai.
String	Int, tandaDecimalnilai. String jika dikonversi keDecimalnilai, dan tandaDecimalnilai dikembalikan. JikaStringtidak dapat dikonversi keDecimal, hasilnya adalahUndefined .
Nilai Lainnya	Undefined .

sin (Desimal)

Mengembalikan sinus angka dalam radian.Decimalargumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: $\text{sin}(0) = 0,0$

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), sinus argumen.
Decimal	Decimal(dengan presisi ganda), sinus argumen.
Boolean	Undefined .

Tipe Argumen	Hasil
String	Decimal, sinus argumen. Jika string tidak dapat dikonversi keDecimal, hasilnya adalahUndefined .
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

sinh (Desimal)

Mengembalikan sinus hiperbolik angka.DecimalNilai dibulatkan menjadi presisi ganda sebelum aplikasi fungsi. Hasilnya adalahDecimalNilai presisi ganda.

Contoh: $\sinh(2.3) = 4.936961805545957$

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), sinus hiperbolik argumen.
Decimal	Decimal(dengan presisi ganda), sinus hiperbolik argumen.
Boolean	Undefined .
String	Decimal, sinus hiperbolik argumen. Jika string tidak dapat dikonversi keDecimal, hasilnya adalahUndefined .
Array	Undefined .
Object	Undefined .

Tipe Argumen	Hasil
Null	Undefined .
Undefined	Undefined .

sqrt (Desimal)

Mengembalikan akar kuadrat dari angka. `Decimal` argumen dibulatkan ke presisi ganda sebelum aplikasi fungsi.

Contoh: `sqrt(9) = 3.010`

Tipe Argumen	Hasil
Int	Akar kuadrat dari argumen.
Decimal	Akar kuadrat dari argumen.
Boolean	Undefined .
String	Akar kuadrat dari argumen. Jika string tidak dapat dikonversi ke <code>Decimal</code> , hasilnya adalah <code>Undefined</code> .
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

tan (Desimal)

Mengembalikan garis singgung angka dalam radian. `Decimal` nilai dibulatkan menjadi presisi ganda sebelum aplikasi fungsi.

Contoh: `tan(3) = -0.1425465430742778`

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), garis singgung argumen.
Decimal	Decimal(dengan presisi ganda), garis singgung argumen.
Boolean	Undefined .
String	Decimal(dengan presisi ganda), garis singgung argumen. Jika string tidak dapat dikonversi keDecimal, hasilnya adalahUndefined .
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

tanh (Desimal)

Mengembalikan tangen hiperbolik angka dalam radian.Decimal nilai dibulatkan menjadi presisi ganda sebelum aplikasi fungsi.

Contoh: $\tanh(2.3) = 0.9800963962661914$

Tipe Argumen	Hasil
Int	Decimal(dengan presisi ganda), garis singgung hiperbolik argumen.
Decimal	Decimal(dengan presisi ganda), garis singgung hiperbolik argumen.
Boolean	Undefined .

Tipe Argumen	Hasil
String	Decimal(dengan presisi ganda), garis singgung hiperbolik argumen. Jika string tidak dapat dikonversi keDecimal, hasilnya adalahUndefined .
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

trunc (Desimal, Integer)

Memotong argumen pertama ke jumlahDecimaltempat yang ditentukan oleh argumen kedua. Jika argumen kedua kurang dari nol, maka akan diatur ke nol. Jika argumen kedua lebih besar dari 34, maka akan diatur ke 34. Trailing nol dilucuti dari hasilnya.

Contoh:

```
trunc(2.3, 0)= 2
```

```
trunc(2.3123, 2)= 2.031
```

```
trunc(2.888, 2)= 2.88
```

```
trunc(2.00, 5)= 2
```

Tipe Argumen 1	Tipe Argumen 2	Hasil
Int	Int	Nilai sumber.
Int / Decimal / String	Int / Decimal	Argumen pertama dipotong dengan panjang yang dijelaskan oleh argumen kedua. Argumen kedua, jika

Tipe Argumen 1	Tipe Argumen 2	Hasil
		bukanInt, akan dibulatkan ke yang terdekatInt. String dikonversi keDecimalvalues. Jika konversi string gagal, hasilnya adalahUndefined .
Nilai Lainnya		Tidak terdefinisi

RunPipelineActivity

Berikut adalah contoh bagaimana Anda akan menggunakan `RunPipelineActivity` perintah untuk menguji aktivitas pipeline. Untuk contoh ini, kami menguji aktivitas matematika.

1. Buat `maths.json` file, yang berisi definisi aktivitas pipeline yang ingin Anda uji.

```
{
  "math": {
    "name": "MyMathActivity",
    "math": "((temp - 32) * 5.0) / 9.0",
    "attribute": "tempC"
  }
}
```

2. Membuat `payloads.json` file, yang berisi contoh payload yang digunakan untuk menguji aktivitas pipeline.

```
[
  "{\"humidity\": 52, \"temp\": 68 }",
  "{\"humidity\": 52, \"temp\": 32 }"
]
```

3. Panggilan `RunPipelineActivities` operasi dari baris perintah.

```
aws iotanalytics run-pipeline-activity --pipeline-activity file://maths.json --
payloads file://payloads.json --cli-binary-format raw-in-base64-out
```

Hal ini menghasilkan hasil berikut.

```
{
  "logResult": "",
  "payloads": [
    "eyJodW1pZG10eSI6NTIsInRlbXAi0jY4LCJ0ZW1wQyI6MjB9",
    "eyJodW1pZG10eSI6NTIsInRlbXAi0jMyLCJ0ZW1wQyI6MH0="
  ]
}
```

Muatan yang tercantum dalam hasil adalah string yang dikodekan sebagai Base64. Ketika string ini diterjemahkan, Anda mendapatkan hasil berikut.

```
{"humidity":52,"temp":68,"tempC":20}
{"humidity":52,"temp":32,"tempC":0}
```

Pesan saluran

AWS IoT Analytics memungkinkan Anda untuk memproses ulang data saluran. Hal ini dapat berguna dalam kasus berikut:

- Anda ingin memutar ulang data tertelan yang ada daripada memulai dari awal.
- Anda membuat update ke pipeline dan ingin membawa data yang ada up-to-date dengan perubahan.
- Anda ingin menyertakan data yang tertelan sebelum Anda membuat perubahan pada opsi penyimpanan yang dikelola pelanggan, izin untuk saluran, atau penyimpanan data.

Parameter

Ketika Anda memproses ulang pesan saluran melalui pipa dengan AWS IoT Analytics, Anda harus menentukan informasi berikut:

`StartPipelineReprocessing`

Mulai memproses ulang pesan saluran melalui pipa.

`ChannelMessages`

Menentukan satu atau lebih set pesan saluran yang ingin Anda proses ulang.

Jika Anda menggunakan `channelMessages` objek, Anda tidak harus menentukan nilai untuk `startTime` dan `endTime`.

`s3Paths`

Menentukan objek Amazon Simple Storage Service (Amazon S3) yang mengidentifikasi objek Amazon Simple Storage Service (Amazon S3). Anda harus menggunakan path lengkap untuk kunci.

Contoh

```
jalat:00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0
```

Jenis: Array string

Batasan anggota Array: 1-100 item.

Kendala panjang: 1-1024 karakter.

endTime

Waktu akhir (eksklusif) dari data saluran yang diproses ulang.

Jika Anda menentukan nilai untuk `endTime` parameter, Anda tidak harus menggunakan `channelMessages` objek.

Jenis: Timestamp

startTime

Waktu mulai (inklusif) data pesan mentah yang diproses ulang.

Jika Anda menentukan nilai untuk `startTime` parameter, Anda tidak harus menggunakan `channelMessages` objek.

Jenis: Timestamp

pipelineName

Nama dari alur untuk memulai pemrosesan ulang.

Jenis: String

Kendala panjang: 1-128 karakter.

Pesan saluran (konsol)

Tutorial ini menunjukkan cara memproses ulang data saluran yang disimpan di objek Amazon S3 yang ditentukan dalam AWS IoT Analytics konsol.

Sebelum memulai, pastikan pesan saluran yang ingin Anda proses ulang disimpan di bucket Amazon S3 yang dikelola pelanggan.

1. Masuk ke [konsol AWS IoT Analytics](#) tersebut.
2. Di panel navigasi, pilih Alur.
3. Pilih jalur target Anda.
4. Pilih Pesan dari Tindakan.
5. Pada Pengolahan ulang alur halaman, pilih Objek S3 untuk Pesan.

Parameter AWS IoT Analytics konsol juga menyediakan pilihan berikut:

- Semua rentang yang tersedia- Memproses ulang semua data yang valid di saluran.

- 120 hari terakhir- Memproses ulang data yang tiba dalam 120 hari terakhir.
 - 90 hari terakhir- Data yang tiba dalam 90 hari terakhir.
 - 30 hari terakhir- Memproses ulang data yang tiba dalam 30 hari terakhir.
 - Rentang kustom- Memproses ulang data yang tiba dalam rentang waktu yang ditentukan. Anda dapat memilih rentang waktu apa pun.
6. Masukkan kunci dari Amazon S3 object yang menyimpan pesan saluran Anda.

Untuk menemukan kunci, lakukan hal berikut:

- a. Pergi ke [Konsol Amazon S3](#).
 - b. Pilih objek Amazon S3 target.
 - c. Di bawah Properti, lihat bagian, salin kuncinya.
7. Pilih Mulai memproses ulang.

Memproses ulang pesan saluran (API)

Saat Anda menggunakan `StartPipelineReprocessingAPI`, perhatikan hal berikut:

- Parameter `startTime` dan `endTime` parameter menentukan kapan data mentah tertelan, tetapi ini adalah perkiraan kasar. Anda dapat putaran ke jam terdekat. Parameter `startTime` inklusif, tapi `endTime` bersifat eksklusif.
- Perintah meluncurkan pemrosesan ulang `asynchronously` dan kembali segera.
- Tidak ada jaminan bahwa pesan yang diproses ulang diproses sesuai urutan yang awalnya diterima. Ini kira-kira sama, tapi tidak tepat.
- Anda dapat membuat hingga 1000 `StartPipelineReprocessing` Permintaan API setiap 24 jam untuk memproses ulang pesan saluran yang sama melalui saluran pipa.
- Memproses ulang data mentah Anda menimbulkan biaya tambahan.

Untuk informasi selengkapnya, lihat [StartPipelineReprocessingAPI](#), di [AWS IoT Analytics Referensi API](#).

Membatalkan aktivitas pemrosesan ulang saluran

Untuk membatalkan aktivitas pemrosesan ulang pipa, gunakan [Batalkan PipelineProcessingAPI](#) atau pilih [Batalkan Pengolahan Ulang pada Aktivitas halaman AWS IoT Analytics konsol](#). Jika Anda

membatalkan pemrosesan ulang, data yang tersisa tidak akan diproses ulang. Anda harus memulai permintaan pemrosesan ulang lainnya.

Gunakan [DescribePipeline](#) API untuk memeriksa status pemrosesan ulang. Lihat `reprocessingSummaries` lapangan dalam respon.

Mengotomatisasi alur kerja

AWS IoT Analytics menyediakan analisis data lanjutan untuk AWS IoT. Anda dapat secara otomatis mengumpulkan data IoT, memprosesnya, menyimpannya, dan menganalisisnya menggunakan analisis data dan alat pembelajaran mesin. Anda dapat menjalankan kontainer yang meng-host kode analitik kustom Anda sendiri atau Notebook Jupyter atau menggunakan kontainer kode kustom pihak ketiga sehingga Anda tidak perlu membuat ulang alat analisis yang ada. Anda dapat menggunakan kemampuan berikut untuk mengambil data input dari penyimpanan data dan memasukkannya ke dalam alur kerja otomatis:

Buat konten set data pada jadwal berulang

Jadwalkan pembuatan konten kumpulan data secara otomatis dengan menentukan pemicu saat Anda menelepon `CreateDataset(triggers:schedule:expression)`. Data yang ada di penyimpanan data digunakan untuk membuat konten dataset. Anda memilih bidang yang Anda inginkan dengan menggunakan kueri SQL (`actions:queryAction:sqlQuery`).

Tentukan interval waktu yang tidak tumpang tindih dan bersebelahan untuk memastikan konten kumpulan data baru hanya berisi data yang telah tiba sejak terakhir kali.

Gunakan `actions:queryAction:filters:deltaTime:offsetSeconds` bidang untuk menentukan interval waktu delta. Kemudian tentukan pemicu untuk membuat konten set data saat interval waktu telah berlalu. Lihat [the section called “Contoh 6 - membuat dataset SQL dengan jendela delta \(CLI\)”](#).

Membuat konten dataset setelah menyelesaikan dataset lain

Memicu pembuatan konten kumpulan data baru saat pembuatan konten kumpulan data lain selesai `triggers:dataset:name`.

Jalankan aplikasi analisis Anda secara otomatis

Kontainer aplikasi analisis data kustom Anda sendiri dan memicu mereka untuk berjalan ketika konten dataset lain dibuat. Dengan cara ini, Anda dapat memberi makan aplikasi Anda dengan data dari konten kumpulan data yang dibuat pada jadwal berulang. Anda dapat secara otomatis mengambil tindakan atas hasil analisis Anda dari dalam aplikasi Anda.

(`actions:containerAction`)

Membuat konten dataset setelah menyelesaikan dataset lain

Memicu pembuatan konten kumpulan data baru saat pembuatan konten kumpulan data lain selesai `triggers:dataset:name`.

Jalankan aplikasi analisis Anda secara otomatis

Kontainer aplikasi analisis data kustom Anda sendiri dan memicu mereka untuk berjalan ketika konten dataset lain dibuat. Dengan cara ini, Anda dapat memberi makan aplikasi Anda dengan data dari konten kumpulan data yang dibuat pada jadwal berulang. Anda dapat secara otomatis mengambil tindakan atas hasil analisis Anda dari dalam aplikasi Anda.

(actions:containerAction)

Kasus penggunaan

Mengotomatiskan pengukuran kualitas produk untuk menurunkan OpEx

Anda memiliki sistem dengan katup pintar yang mengukur tekanan, kelembaban, dan suhu. Sistem menyusun peristiwa secara berkala dan juga ketika peristiwa tertentu terjadi, seperti ketika nilai terbuka dan ditutup. Dengan AWS IoT Analytics, Anda dapat mengotomatiskan analisis yang mengumpulkan data yang tidak tumpang tindih dari jendela periodik ini dan membuat laporan KPI tentang kualitas produk akhir. Setelah memproses setiap batch, Anda mengukur kualitas produk secara keseluruhan dan menurunkan pengeluaran operasional Anda melalui volume run yang dimaksimalkan.

Mengotomatiskan analisis armada perangkat

Anda menjalankan analitik (algoritme, ilmu data, atau ML untuk KPI) setiap 15 menit pada data yang dihasilkan oleh 100 perangkat. Dengan setiap siklus analisis menghasilkan dan menyimpan status untuk analisis berikutnya dijalankan. Untuk setiap analisis Anda, Anda hanya ingin menggunakan data yang diterima dalam jendela waktu tertentu. Dengan AWS IoT Analytics Anda dapat mengatur analisis Anda dan membuat KPI dan melaporkan untuk setiap run kemudian menyimpan data untuk analisis future.

Deteksi otomatis anomali

AWS IoT Analytics memungkinkan Anda untuk mengotomatiskan alur kerja deteksi anomali Anda yang harus Anda jalankan secara manual setiap 15 menit pada data baru yang telah tiba di penyimpanan data. Anda juga dapat mengotomatiskan dasbor yang menunjukkan penggunaan perangkat dan pengguna teratas dalam jangka waktu tertentu.

Memprediksi hasil proses industri

Anda memiliki jalur produksi industri. Menggunakan data yang dikirim ke AWS IoT Analytics, termasuk pengukuran proses yang tersedia, Anda dapat mengoperasionalkan alur kerja analitis untuk memprediksi hasil proses. Data untuk model dapat diatur dalam matriks $M \times N$ di mana

setiap baris berisi data dari berbagai titik waktu di mana sampel laboratorium diambil. AWS IoT Analytics membantu Anda mengoperasionalkan alur kerja analitis Anda dengan membuat jendela delta dan menggunakan alat sains data Anda untuk membuat KPI dan menyimpan keadaan perangkat pengukuran.

Menggunakan kontainer Docker

Bagian ini mencakup informasi tentang cara membuat container Docker Anda sendiri. Ada risiko keamanan jika Anda menggunakan kembali kontainer Docker yang dibuat oleh pihak ketiga: kontainer ini dapat mengeksekusi kode arbitrer dengan izin pengguna Anda. Pastikan Anda mempercayai penulis wadah pihak ketiga apa pun sebelum menggunakannya.

Berikut adalah langkah-langkah yang akan Anda ambil untuk mengatur analisis data periodik pada data yang telah tiba sejak analisis terakhir dilakukan:

1. Buat container Docker yang berisi aplikasi data Anda ditambah pustaka yang diperlukan atau dependensi lainnya.

Klaster IoT Analytics Ekstensi Jupyter menyediakan API containerization untuk membantu proses containerization. Anda juga dapat menjalankan gambar kreasi Anda sendiri di mana Anda membuat atau merakit toolset aplikasi Anda untuk melakukan analisis data yang diinginkan atau perhitungan. AWS IoT Analytics memungkinkan Anda untuk menentukan sumber data input ke aplikasi kemas dan tujuan untuk data keluaran wadah Docker melalui variabel. ([Variabel Input/Output kontainer Docker kustom](#) berisi informasi lebih lanjut tentang menggunakan variabel dengan wadah khusus.)

2. Unggah kontainer ke [Amazon ECR](#) registri.
3. Buat penyimpanan data untuk menerima dan menyimpan pesan (data) dari perangkat (iotanalytics: [CreateDatastore](#))
4. Buat saluran tempat pesan dikirim (iotanalytics: [CreateChannel](#)).
5. Buat pipeline untuk menghubungkan saluran ke penyimpanan data (iotanalytics: [CreatePipeline](#)).
6. Buat IAM role yang memberikan izin untuk mengirim data pesan ke AWS IoT Analytics saluran (iam: [CreateRole](#).)
7. Buat aturan IoT yang menggunakan kueri SQL untuk menghubungkan saluran ke sumber data pesan (iot: [CreateTopicRule](#) bidang `topicRulePayload:actions:iotAnalytics`). Ketika perangkat mengirim pesan dengan visa topik yang sesuai MQTT, itu dialihkan ke saluran

Anda. Atau, Anda dapat menggunakan `iotanalytics: BatchPutMessage` untuk mengirim pesan langsung ke saluran dari perangkat yang mampu menggunakan `AWSSDK` atau `AWS CLI`.

8. Buat dataset SQL yang pembuatannya dipicu oleh jadwal waktu (`iotanalytics: CreateDataset`, bidang `actions: queryAction: sqlQuery`).

Anda juga menentukan pra-filter yang akan diterapkan ke data pesan untuk membantu membatasi pesan ke pesan yang telah tiba sejak eksekusi terakhir tindakan.

(Bidang `actions: queryAction: filters: deltaTime: timeExpression` memberikan ekspresi di mana waktu pesan mungkin ditentukan. sementara bidang `actions: queryAction: filters: deltaTime: offsetSeconds` menentukan kemungkinan latensi dalam kedatangan pesan.)

Pra-filter, bersama dengan jadwal pemacu, menentukan jendela delta Anda. Setiap dataset SQL baru dibuat menggunakan pesan yang diterima sejak terakhir kali dataset SQL dibuat. (Bagaimana dengan pertama kali dataset SQL dibuat? Perkiraan kapan terakhir kali dataset akan dibuat berdasarkan jadwal dan pra-filter.)

9. Buat dataset lain yang dipicu oleh pembuatan yang pertama (`CreateDataset` bidang `trigger: dataset`). Untuk kumpulan data ini, Anda menentukan tindakan kontainer (diajukan `actions: containerAction`) yang menunjuk ke, dan memberikan informasi yang diperlukan untuk menjalankan, wadah Docker yang Anda buat pada langkah pertama. Di sini Anda juga menentukan:

- ARN dari kontainer docker disimpan dalam akun Anda (`image`).
- ARN dari peran yang memberikan izin kepada sistem untuk mengakses sumber daya yang diperlukan untuk menjalankan tindakan kontainer (`executionRoleArn`).
- Konfigurasi sumber daya yang menjalankan aksi kontainer (`resourceConfiguration`).
- Jenis jika sumber daya komputasi yang digunakan untuk menjalankan aksi kontainer (`computeType` nilai yang mungkin: `ACU_1 [vCPU=4, memory=16GiB]` or `ACU_2 [vCPU=8, memory=32GiB]`).
- Ukuran (GB) penyimpanan tetap tersedia untuk instans sumber daya yang digunakan untuk menjalankan aksi kontainer (`volumeSizeInGB`).
- Nilai-nilai variabel yang digunakan dalam konteks eksekusi aplikasi (pada dasarnya, parameter diloloskan ke aplikasi) (`variables`).

Variabel-variabel ini diganti pada saat sebuah wadah dijalankan. Hal ini memungkinkan Anda untuk menjalankan wadah yang sama dengan variabel yang berbeda (parameter)

yang disediakan pada saat konten dataset dibuat. Kluster IoT Analytics Ekstensi Jupyter menyederhanakan proses ini dengan secara otomatis mengenali variabel dalam buku catatan dan membuatnya tersedia sebagai bagian dari proses containerization. Anda dapat memilih variabel yang diakui atau menambahkan variabel kustom Anda sendiri. Sebelum menjalankan wadah, sistem menggantikan masing-masing variabel ini dengan nilai saat ini pada saat eksekusi.

- Salah satu variabelnya adalah nama kumpulan data yang konten terbarunya digunakan sebagai input ke aplikasi (ini adalah nama kumpulan data yang Anda buat di langkah sebelumnya) (`datasetContentVersionValue:datasetName`).

Dengan kueri SQL dan jendela delta untuk menghasilkan dataset, dan wadah dengan aplikasi Anda, AWS IoT Analytics membuat dataset produksi terjadwal yang berjalan pada interval yang Anda tentukan pada data dari jendela delta, menghasilkan output yang Anda inginkan dan mengirim pemberitahuan.

Anda dapat menjeda aplikasi set data produksi Anda dan melanjutkannya kapan pun Anda memilih untuk melakukannya. Ketika Anda melanjutkan aplikasi dataset produksi Anda, AWS IoT Analytics, secara default, menangkap semua data yang telah tiba sejak eksekusi terakhir, tetapi belum dianalisis. Anda juga dapat mengkonfigurasi bagaimana Anda ingin melanjutkan panjang jendela pekerjaan dataset produksi Anda) dengan melakukan serangkaian berjalan berturut-turut. Atau, Anda dapat melanjutkan aplikasi dataset produksi dengan hanya menangkap data yang baru tiba yang sesuai dengan ukuran jendela delta Anda yang ditentukan.

Harap perhatikan batasan berikut saat membuat atau mendefinisikan dataset yang dipicu oleh pembuatan dataset lain:

- Hanya dataset kontainer yang dapat dipicu oleh kumpulan data SQL.
- Sebuah dataset SQL dapat memicu paling banyak 10 dataset kontainer.

Kesalahan berikut dapat dikembalikan saat membuat dataset kontainer yang dipicu oleh dataset SQL:

- “Memicu dataset hanya dapat ditambahkan pada dataset kontainer”
- “Hanya ada satu set data pemicu”

Kesalahan ini terjadi jika Anda mencoba untuk menentukan dataset kontainer yang dipicu oleh dua dataset SQL yang berbeda.

- “Kumpulan data pemicu <dataset-name>tidak dapat dipicu oleh kumpulan data kontainer”

Kesalahan ini terjadi jika Anda mencoba untuk menentukan dataset kontainer lain yang dipicu oleh dataset kontainer lain.

- “<N>dataset sudah tergantung pada <dataset-name>dataset.”

Kesalahan ini terjadi jika Anda mencoba untuk menentukan dataset kontainer lain yang dipicu oleh dataset SQL yang sudah memicu 10 dataset kontainer.

- “Tepat satu jenis pemicu harus disediakan”

Kesalahan ini terjadi adalah Anda mencoba untuk menentukan dataset yang dipicu oleh kedua pemicu jadwal dan pemicu dataset.

Variabel input/output kontainer Docker kustom

Bagian ini menunjukkan bagaimana program yang dijalankan oleh gambar Docker kustom Anda dapat membaca variabel masukan dan meng-upload output.

Berkas parameter

Variabel input dan tujuan yang ingin Anda unggah output disimpan dalam file JSON yang terletak di `opt/ml/input/data/iotanalytics/param` pada instance yang mengeksekusi image docker Anda. Berikut ini adalah contoh isi file tersebut.

```
{
  "Context": {
    "OutputUri": {
      "html": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/iotanalytics-xxxxxxx/output.html",
      "ipynb": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/iotanalytics-xxxxxxx/output.ipynb"
    }
  },
  "Variables": {
    "source_dataset_name": "mydataset",
    "source_dataset_version_id": "xxxx",
    "example_var": "hello world!",
    "custom_output": "s3://aws-iot-analytics/dataset-xxxxxxx/notebook/results/iotanalytics-xxxxxxx/output.txt"
  }
}
```

```
}

```

Selain nama dan ID versi set data Anda, `Variables` bagian berisi variabel yang ditentukan dalam `iotanalytics:CreateDatasetInvocation`— dalam contoh ini, variabel `example_var` diberi nilai `hello world!`. URI keluaran khusus juga disediakan di `custom_output` Variabel. `KlasterOutputUri` bidang berisi lokasi default yang kontainer dapat meng-upload output— dalam contoh ini, URI keluaran default disediakan untuk kedua `ipynb` dan `html` output.

Variabel input

Program yang diluncurkan oleh gambar Docker Anda dapat membaca variabel dari `params` berkas. Berikut ini adalah contoh program yang membuka program contoh program yang membuka program `params` file, mem-parsing, dan mencetak nilai `example_var` Variabel.

```
import json

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
    example_var = params["Variables"]["example_var"]
    print(example_var)
```

Output

Program yang diluncurkan oleh image Docker Anda mungkin juga menyimpan outputnya di lokasi Amazon S3. Output harus dimuat dengan `"bucket-owner-full-control"` [daftar kontrol akses](#). Daftar akses memberikan AWS IoT Analytics kontrol layanan atas output upload. Dalam contoh ini kita memperpanjang yang sebelumnya untuk meng-upload isi `example_var` ke lokasi Amazon S3 yang ditentukan oleh `custom_output` di dalam `params` berkas.

```
import boto3
import json
from urllib.parse import urlparse

ACCESS_CONTROL_LIST = "bucket-owner-full-control"

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
    example_var = params["Variables"]["example_var"]

    outputUri = params["Variables"]["custom_output"]
```

```
# break the S3 path into a bucket and key
bucket = urlparse(outputUri).netloc
key = urlparse(outputUri).path.lstrip("/")

s3_client = boto3.client("s3")
s3_client.put_object(Bucket=bucket, Key=key, Body=example_var, ACL=ACCESS_CONTROL_LIST)
```

Izin

Anda harus membuat dua peran. Satu peran memberikan izin untuk meluncurkan SageMaker misalnya dalam rangka untuk containerize notebook. Peran lain diperlukan untuk mengeksekusi wadah.

Anda dapat membuat peran pertama secara otomatis atau manual. Jika Anda membuat yang baru SageMaker misalnya dengan AWS IoT Analytics konsol, Anda diberi opsi untuk secara otomatis membuat peran baru yang memberikan semua hak istimewa yang diperlukan untuk mengeksekusi SageMaker contoh dan containerize notebook. Atau, Anda dapat membuat peran dengan hak istimewa ini secara manual. Untuk melakukan ini, buat peran dengan `AmazonSageMakerFullAccess` kebijakan dilampirkan dan tambahkan kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:BatchGetImage",
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:DescribeRepositories",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```

    "s3:GetObject"
  ],
  "Resource": "arn:aws:s3:::iotanalytics-notebook-containers/*"
}
]
}

```

Anda harus secara manual membuat peran kedua yang memberikan izin untuk mengeksekusi kontainer. Anda harus melakukan ini bahkan jika Anda menggunakan AWS IoT Analytics konsol untuk membuat peran pertama secara otomatis. Buat peran dengan kebijakan berikut dan kebijakan kepercayaan terlampir.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:PutObject",
        "s3:GetObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::aws-*-dataset-*/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",

```

```

        "logs:PutLogEvents"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListBucket",
      "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
  }
]
}

```

Berikut ini adalah contoh kebijakan.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": ["sagemaker.amazonaws.com", "iotanalytics.amazonaws.com"]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Menggunakan CreateDataset API melalui Java dan AWS CLI

Membuat set data. Sebuah dataset menyimpan data yang diambil dari penyimpanan data dengan menerapkan `queryAction` (query SQL) atau `containerAction` (mengeksekusi aplikasi kemas). Operasi ini menciptakan kerangka dataset. Dataset dapat diisi secara manual dengan memanggil `CreateDatasetContent` atau secara otomatis sesuai dengan `trigger` Anda tentukan. Untuk informasi selengkapnya, lihat [CreateDataset](#) dan [CreateDatasetContent](#).

Topik

- [Contoh 1 - membuat dataset SQL \(java\)](#)
- [Contoh 2 - membuat dataset SQL dengan jendela delta \(java\)](#)
- [Contoh 3 - membuat dataset kontainer dengan pemicu jadwal sendiri \(java\)](#)
- [Contoh 4 - membuat dataset kontainer dengan dataset SQL sebagai pemicu \(java\)](#)
- [Contoh 5 - membuat dataset SQL \(CLI\)](#)
- [Contoh 6 - membuat dataset SQL dengan jendela delta \(CLI\)](#)

Contoh 1 - membuat dataset SQL (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("SQLAction1");
action.setQueryAction(new SqlQueryDatasetAction().withSqlQuery("select * from
  DataStoreName"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Output pada kesuksesan:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>, RetentionPeriod: {unlimited:  
true} or {numberOfDays: 10, unlimited: false}}
```

Contoh 2 - membuat dataset SQL dengan jendela delta (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();  
request.setDatasetName(dataSetName);  
DatasetAction action = new DatasetAction();  
  
//Create Filter for DeltaTime  
QueryFilter deltaTimeFilter = new QueryFilter();  
deltaTimeFilter.withDeltaTime(  
    new DeltaTime()  
        .withOffsetSeconds(-1 * EstimatedDataDelayInSeconds)  
        .withTimeExpression("from_unixtime(timestamp)"));  
  
//Create Action  
action.setActionName("SQLActionWithDeltaTime");  
action.setQueryAction(new SqlQueryDatasetAction()  
    .withSqlQuery("SELECT * from DataStoreName")  
    .withFilters(deltaTimeFilter));  
  
// Add Action to Actions List  
List<DatasetAction> actions = new ArrayList<DatasetAction>();  
actions.add(action);  
  
//Create Trigger  
DatasetTrigger trigger = new DatasetTrigger();  
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));  
  
//Add Trigger to Triggers List  
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();  
triggers.add(trigger);  
  
// Add Triggers and Actions to CreateDatasetRequest object  
request.setActions(actions);  
request.setTriggers(triggers);  
  
// Add RetentionPeriod to CreateDatasetRequest object  
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));  
final CreateDatasetResult result = iot.createDataset(request);
```

Output pada kesuksesan:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>, RetentionPeriod: {unlimited: true} or {numberOfDays: 10, unlimited: false}}
```

Contoh 3 - membuat dataset kontainer dengan pemicu jadwal sendiri (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
    .withImage(ImageURI)
    .withExecutionRoleArn(ExecutionRoleArn)
    .withResourceConfiguration(
        new ResourceConfiguration()
            .withComputeType(new ComputeType().withAcu(1))
            .withVolumeSizeInGB(1))
    .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
```

```
final CreateDatasetResult result = iot.createDataset(request);
```

Output pada kesuksesan:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>, RetentionPeriod: {unlimited: true} or {numberOfDays: 10, unlimited: false}}
```

Contoh 4 - membuat dataset kontainer dengan dataset SQL sebagai pemicu (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
    .withImage(ImageURI)
    .withExecutionRoleArn(ExecutionRoleArn)
    .withResourceConfiguration(
        new ResourceConfiguration()
            .withComputeType(new ComputeType().withAcu(1))
            .withVolumeSizeInGB(1))
    .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger()
    .withDataset(new TriggeringDataset()
        .withName(TriggeringSQLDataSetName));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
```

```
request.setTriggers(triggers);
final CreateDatasetResult result = iot.createDataset(request);
```

Output pada kesuksesan:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>}
```

Contoh 5 - membuat dataset SQL (CLI)

```
aws iotanalytics --endpoint <EndPoint> --region <Region> create-dataset --dataset-name="<datasetName>" --actions="[{"actionName":"<ActionName>", "queryAction":{"sqlQuery":"<SQLQuery>"} }]" --retentionPeriod numberOfDays=10
```

Output pada kesuksesan:

```
{
  "datasetName": "<datasetName>",
  "datasetArn": "<datasetARN>",
  "retentionPeriod": {unlimited: true} or {numberOfDays: 10, unlimited: false}
}
```

Contoh 6 - membuat dataset SQL dengan jendela delta (CLI)

Jendela Delta adalah serangkaian interval waktu yang ditentukan pengguna, tidak tumpang tindih, dan berkelanjutan. Delta windows memungkinkan Anda untuk membuat konten set data dengan, dan melakukan analisis, data baru yang telah tiba di penyimpanan data sejak analisis terakhir. Anda membuat jendela delta dengan mengatur `deltaTime` dalam `filters` dari `queryAction` set data ([CreateDataset](#)). Biasanya, Anda akan ingin membuat konten dataset secara otomatis dengan juga mengatur pemicu interval waktu (`triggers:schedule:expression`). Pada dasarnya, ini memungkinkan Anda untuk memfilter pesan yang telah tiba selama jendela waktu tertentu, sehingga data yang terkandung dalam pesan dari jendela waktu sebelumnya tidak dihitung dua kali.

Dalam contoh ini, kita membuat dataset baru yang secara otomatis membuat konten dataset baru setiap 15 menit hanya menggunakan data yang telah tiba sejak terakhir kali. Kami menentukan 3 menit (180 detik) `deltaTimeoffset` yang memungkinkan penundaan 3 menit agar pesan tiba di penyimpanan data yang ditentukan. Jadi, jika konten dataset dibuat pada pukul 10:30 pagi, data yang digunakan (termasuk dalam konten dataset) adalah bahwa dengan cap waktu antara 10:12 AM dan 10:27 AM (yaitu 10:30 AM - 15 menit - 3 menit untuk 10:30 AM - 3 menit).

```
aws iotanalytics --endpoint <EndPoint> --region <Region> create-dataset --cli-input-  
json file://delta-window.json
```

Dimana file `delta-window.json` berisi berikut.

```
{  
  "datasetName": "delta_window_example",  
  "actions": [  
    {  
      "actionName": "delta_window_action",  
      "queryAction": {  
        "sqlQuery": "SELECT temperature, humidity, timestamp FROM my_datastore",  
        "filters": [  
          {  
            "deltaTime": {  
              "offsetSeconds": -180,  
              "timeExpression": "from_unixtime(timestamp)"  
            }  
          }  
        ]  
      }  
    }  
  ],  
  "triggers": [  
    {  
      "schedule": {  
        "expression": "cron(0/15 * * * ? *)"  
      }  
    }  
  ]  
}
```

Output pada kesuksesan:

```
{  
  "datasetName": "<datasetName>",  
  "datasetArn": "<datasetARN>",  
}
```

Kontainerisasi notebook

Bagian ini mencakup informasi tentang cara membuat kontainer Docker menggunakan notebook Jupyter. Ada risiko keamanan jika Anda menggunakan kembali notebook yang dibuat oleh pihak ketiga: kontainer yang disertakan dapat mengeksekusi kode arbitrer dengan izin pengguna Anda. Selain itu, HTML yang dihasilkan oleh notebook dapat ditampilkan di AWS IoT Analytics konsol, menyediakan vektor serangan potensi pada komputer menampilkan HTML. Pastikan Anda mempercayai penulis notebook pihak ketiga sebelum menggunakannya.

Salah satu opsi untuk melakukan fungsi analitis lanjutan adalah dengan menggunakan [Notebook Jupyter](#). Jupyter Notebook menyediakan alat ilmu data yang kuat yang dapat melakukan pembelajaran mesin dan berbagai analisis statistik. Untuk informasi selengkapnya, lihat [Templat notebook](#). (Perhatikan bahwa saat ini kami tidak mendukung kontainerisasi di dalamnya JupyterLab.) Anda dapat mengemas Notebook dan pustaka Jupyter Anda ke dalam wadah yang secara berkala berjalan pada kumpulan data baru saat diterima oleh AWS IoT Analytics selama jendela waktu delta yang Anda tentukan. Anda dapat menjadwalkan pekerjaan analisis yang menggunakan container dan data tersegmentasi baru yang diambil dalam jendela waktu yang ditentukan, lalu menyimpan output pekerjaan untuk analitik terjadwal di future.

Jika Anda telah membuat SageMaker Instance menggunakan AWS IoT Analytics konsol setelah 23 Agustus 2018, maka instalasi ekstensi containerization telah dilakukan untuk Anda secara otomatis [dan Anda dapat mulai membuat gambar dalam kontainer](#). Jika tidak, ikuti langkah-langkah yang tercantum dalam bagian ini untuk mengaktifkan containerization di SageMaker misalnya. Berikut ini, Anda memodifikasi SageMaker Peran Eksekusi untuk memungkinkan Anda mengunggah gambar kontainer ke Amazon EC2 dan Anda menginstal ekstensi kontainerisasi.

Aktifkan kontainerisasi instance notebook yang tidak dibuat melalui AWS IoT Analytics konsol

Kami sarankan Anda membuat baru SageMaker contoh melalui AWS IoT Analytics konsol bukannya mengikuti langkah-langkah ini. Instans baru secara otomatis mendukung containerization.

Jika Anda memulai ulang SageMaker misalnya setelah mengaktifkan containerization seperti yang ditunjukkan di sini, Anda tidak perlu menambahkan kembali peran dan kebijakan IAM, tetapi Anda harus menginstal ulang ekstensi, seperti yang ditunjukkan pada langkah terakhir.

1. Untuk memberikan akses instans notebook Anda ke Amazon ECS, pilih SageMaker contoh pada SageMaker halaman:

Amazon SageMaker

Dashboard

▼ Notebook

 Notebook instances

 Lifecycle configurations

▼ Training

 Training jobs

Amazon SageMaker > Notebook instances

Notebook instances

Open Start Update settings Actions

Search notebook instances

Name	Instance	Creation time
exampleNotebookInstance	ml.t2.medium	Jul 03, 2018 21:25 UTC

2. Di bawah ARN IAM role Pilih SageMaker Peran eksekusi.

Amazon SageMaker

Dashboard

▼ Notebook

 Notebook instances

 Lifecycle configurations

▼ Training

 Training jobs

 Hyperparameter tuning jobs

▼ Inference

 Models

 Endpoint configurations

 Endpoints

Amazon SageMaker > Notebook instances > exampleNotebookInstance

exampleNotebookInstance

Delete Stop Start Open

Notebook instance settings

Edit

Name	exampleNotebookInstance	Notebook instance type	ml.t2.medium
ARN	arn:aws:sagemaker:us-east-1:██████████:notebook-instance/exampleNotebookInstance	Storage	5GB EBS
Lifecycle configuration	—	Encryption key	
Status	⌚ Pending	IAM role ARN	arn:aws:iam:██████████:role/service-role/AmazonSageMaker-ExecutionRole-20180620T141485

3. Pilih Lampirkan kebijakan, lalu tentukan dan lampirkan kebijakan yang ditunjukkan di [izin](#). Jika AmazonSageMakerFullAccess kebijakan belum terpasang, lampirkan juga.

Permissions Trust relationships Access Advisor Revoke sessions

Attach policy

Attached policies: 7

Anda juga harus mengunduh kode containerization dari Amazon S3 dan menginstalnya pada instance notebook Anda, Langkah pertama adalah mengakses SageMaker terminal contoh.

1. Di dalam Jupyter, pilih Baru.

jupyter

Quit

Files Running Clusters SageMaker Examples Conda



Upload New ↕

- Di menu yang muncul, pilih Terminal.



- Di dalam terminal, masukkan perintah berikut untuk mengunduh kode, menginstalnya, dan menginstalnya. Perhatikan bahwa perintah ini membunuh setiap proses yang dijalankan oleh notebook Anda pada ini SageMaker misalnya.

jupyter

sh-4.2\$ █

```
cd /tmp
```

```
aws s3 cp s3://iotanalytics-notebook-containers/iota_notebook_containers.zip /tmp
```

```
unzip iota_notebook_containers.zip
```

```
cd iota_notebook_containers
```

```
chmod u+x install.sh
```

```
./install.sh
```

Tunggu satu atau dua menit hingga ekstensi divalidasi dan diinstal.

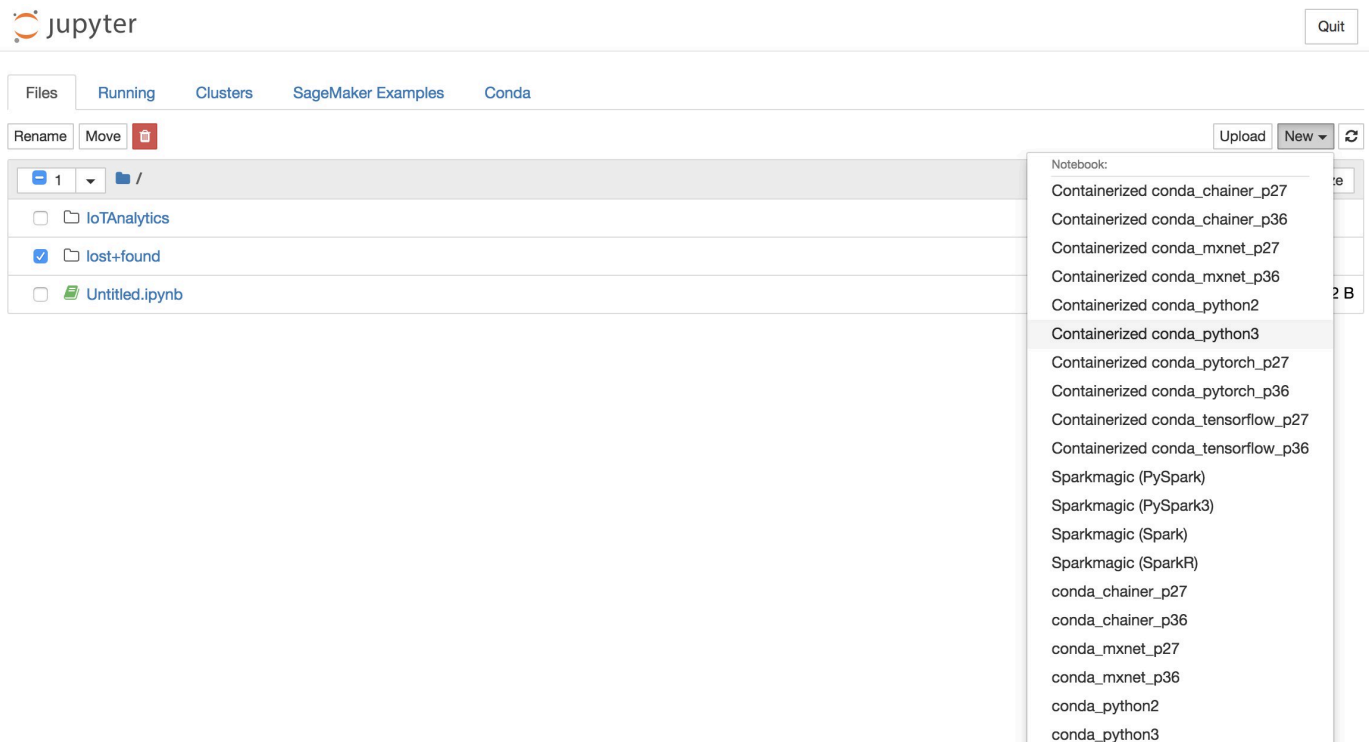
Memperbarui ekstensi kontainerisasi notebook

Jika Anda membuat SageMaker Instance melalui AWS IoT Analytics konsol setelah 23 Agustus 2018, maka ekstensi kontainerisasi diinstal secara otomatis. Anda dapat memperbarui ekstensi dengan memulai ulang instans Anda dari SageMaker Konsol. Jika Anda menginstal ekstensi secara manual, maka Anda dapat memperbaruinya dengan menjalankan kembali perintah terminal yang tercantum dalam Aktifkan Containerization Of Notebook Instans Not Created Via AWS IoT Analytics Konsol.

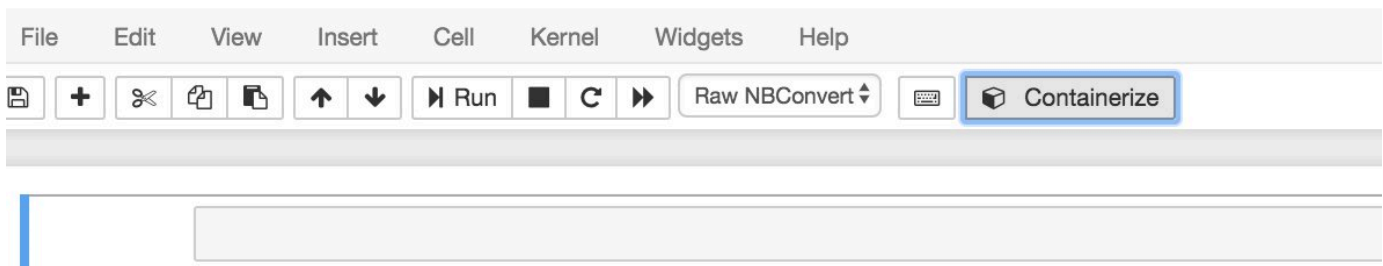
Membuat citra kontainerisasi

Pada bagian ini kami menunjukkan langkah-langkah yang diperlukan untuk membuat wadah notebook. Untuk memulai, buka Notebook Jupyter Anda untuk membuat notebook dengan kernel kemas.

1. Di Notebook Jupyter Anda, pilih **Baru**, lalu pilih jenis kernel yang Anda inginkan dari daftar dropdown. (Jenis kernel harus dimulai dengan "Containerized" dan diakhiri dengan kernel apa pun yang akan Anda pilih. Misalnya, jika Anda hanya menginginkan lingkungan Python 3.0 biasa seperti "conda_python3", pilih "Containerized conda_python3").



2. Setelah Anda menyelesaikan pekerjaan pada notebook Anda dan Anda ingin memasangnya, pilih Kontainerisasi.



3. Masukkan nama untuk notebook kontainer. Anda juga dapat memasukkan deskripsi opsional.

1. Name
2. Input Variables
3. Select AWS ECR Repository
4. Review
5. Monitor Progress

Container Name *

Container Description

Next

Exit

4. Tentukan Variabel Input (parameter) bahwa notebook Anda harus dipanggil dengan. Anda dapat memilih variabel input yang secara otomatis terdeteksi dari notebook Anda atau menentukan variabel kustom. (Perhatikan bahwa variabel input hanya terdeteksi jika Anda sebelumnya telah mengeksekusi notebook Anda.) Untuk setiap variabel masukan memilih jenis. Anda juga dapat memasukkan deskripsi opsional variabel input.

1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Name	Type	Description	
<input type="text" value="ounces"/>	<input type="text" value="Double"/>	<input type="text"/>	<input type="button" value="X"/>
<input type="text" value="brand"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="button" value="X"/>

Showing 1 to 2 of 2 variables

Previous Next

5. Pilih repositori Amazon ECR di mana gambar yang dibuat dari notebook harus diunggah.

1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Please upload different notebooks to different repositories.

Repository Name Create Search:

Name
my-repo
my-repo2
my-repo3

Showing 1 to 3 of 3 repositories Previous Next

6. Pilih Kontainerisasi untuk memulai proses.

Anda akan disajikan dengan ikhtisar meringkas masukan Anda. Perhatikan bahwa setelah memulai proses, Anda tidak dapat membatalkannya. Prosesnya dapat berlangsung hingga satu jam.

1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Container Name: Beer-Tastiness-Calculator**Container Description:****Upload To:** my-repo

Variable Name	Type	Description
ounces	Double	
brand	String	

Showing 1 to 2 of 2 variables

Previous

1

Next

Previous

Containerize

Exit

7. Halaman berikutnya menunjukkan kemajuan.

1. Name

2. Input Variables

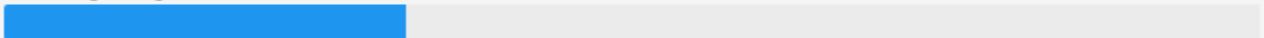
3. Select AWS ECR Repository

4. Review

5. Monitor Progress

The containerization process typically completes within 30 minutes.

Creating Image...



Exit

8. Jika Anda tidak sengaja menutup browser Anda, Anda dapat memantau status proses containerisasi dari Notebook bagian AWS IoT Analytics konsol.
9. Setelah proses selesai, gambar dalam peti kemas disimpan di Amazon ECR yang siap digunakan.

Containerize Notebook



1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Creating Image...

Uploading Image...

You can now use this notebook for scheduled analysis of your Data Sets.

[Go To Data Sets](#)[Exit](#)

Menggunakan wadah khusus untuk analisis

Bagian ini mencakup informasi tentang cara membuat kontainer Docker menggunakan notebook Jupyter. Ada risiko keamanan jika Anda menggunakan kembali notebook yang dibuat oleh pihak ketiga: kontainer yang disertakan dapat mengeksekusi kode arbitrer dengan izin pengguna Anda. Selain itu, HTML yang dihasilkan oleh notebook dapat ditampilkan di AWS IoT Analytics konsol, menyediakan vektor serangan potensi pada komputer menampilkan HTML. Pastikan Anda mempercayai penulis notebook pihak ketiga sebelum menggunakannya.

Anda dapat membuat wadah khusus Anda sendiri dan menjalankannya dengan AWS IoT Analytics layanan. Untuk melakukannya, Anda menyiapkan image Docker dan mengunggahnya ke Amazon ECR, lalu menyiapkan kumpulan data untuk menjalankan aksi kontainer. Bagian ini memberikan contoh proses menggunakan Oktaf.

Tutorial ini mengasumsikan bahwa Anda memiliki:

- Octave diinstal di komputer lokal Anda

- Akun Docker yang diatur di komputer lokal Anda
- SesiAWSakun Amazon ECR atauAWS IoT Analyticsakses

Langkah 1: Menyiapkan citra Docker

Ada tiga file utama yang Anda butuhkan untuk tutorial ini. Nama dan isinya ada di sini:

- **Dockerfile**— Penyiapan awal untuk proses containerization Docker.

```
FROM ubuntu:16.04

# Get required set of software
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN apt-get install -y octave
RUN apt-get install -y python3-pip

# Get boto3 for S3 and other libraries
RUN pip3 install --upgrade pip
RUN pip3 install boto3
RUN pip3 install urllib3

# Move scripts over
ADD moment moment
ADD run-octave.py run-octave.py

# Start python script
ENTRYPOINT ["python3", "run-octave.py"]
```

- **run-octave.py**- Mem-parsing JSON dariAWS IoT Analytics, menjalankan skrip Octave, dan mengunggah artefak ke Amazon S3.

```
import boto3
import json
import os
import sys
from urllib.parse import urlparse

# Parse the JSON from IoT Analytics
with open('/opt/ml/input/data/iotanalytics/params') as params_file:
    params = json.load(params_file)
```

```

variables = params['Variables']

order = variables['order']
input_s3_bucket = variables['inputDataS3BucketName']
input_s3_key = variables['inputDataS3Key']
output_s3_uri = variables['octaveResultS3URI']

local_input_filename = "input.txt"
local_output_filename = "output.mat"

# Pull input data from S3...
s3 = boto3.resource('s3')
s3.Bucket(input_s3_bucket).download_file(input_s3_key, local_input_filename)

# Run Octave Script
os.system("octave moment {} {} {}".format(local_input_filename,
    local_output_filename, order))

# # Upload the artifacts to S3
output_s3_url = urlparse(output_s3_uri)
output_s3_bucket = output_s3_url.netloc
output_s3_key = output_s3_url.path[1:]

s3.Object(output_s3_bucket, output_s3_key).put(Body=open(local_output_filename,
    'rb'), ACL='bucket-owner-full-control')

```

- **moment-** Skrip Oktaf sederhana yang menghitung momen berdasarkan file input atau output dan urutan tertentu.

```

#!/usr/bin/octave -qf

arg_list = argv ();
input_filename = arg_list{1};
output_filename = arg_list{2};
order = str2num(arg_list{3});

[D,delimiterOut]=importdata(input_filename)
M = moment(D, order)

save(output_filename, 'M')

```

1. Unduh isi setiap file. Buat direktori baru dan tempatkan semua file di dalamnya dan kemudiandcke direktori itu.
2. Jalankan perintah berikut.

```
docker build -t octave-moment .
```

3. Anda akan melihat gambar baru di repositori Docker Anda. Verifikasi dengan menjalankan perintah berikut.

```
docker image ls | grep octave-moment
```

Langkah 2: Unggah gambar Docker ke repositori Amazon ECR

1. Buat repositori di Amazon ECR.

```
aws ecr create-repository --repository-name octave-moment
```

2. Dapatkan login ke lingkungan Docker Anda.

```
aws ecr get-login
```

3. Salin output dan jalankan. Outputnya akan terlihat seperti berikut.

```
docker login -u AWS -p password -e none https://your-aws-account-id.dkr.ecr..amazonaws.com
```

4. Tandai gambar yang Anda buat dengan tag repositori Amazon ECR.

```
docker tag your-image-id your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

5. Dorong citra ke Amazon ECR.

```
docker push your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

Langkah 3: Unggah data sampel Anda ke bucket Amazon S3

1. Unduh file berikut `input.txt`.

```

0.857549 -0.987565 -0.467288 -0.252233 -2.298007
0.030077 -1.243324 -0.692745 0.563276 0.772901
-0.508862 -0.404303 -1.363477 -1.812281 -0.296744
-0.203897 0.746533 0.048276 0.075284 0.125395
0.829358 1.246402 -1.310275 -2.737117 0.024629
1.206120 0.895101 1.075549 1.897416 1.383577

```

2. Buat bucket Amazon S3 yang disebut `octave-sample-data-your-aws-account-id`.
3. Pengunggahan file `input.txt` ke bucket Amazon S3 yang baru saja Anda buat. Seharusnya sekarang Anda memiliki ember bernama `octave-sample-data-your-aws-account-id` yang berisi `input.txt` berkas.

Langkah 4: Untuk membuat peran eksekusi kontainer

1. Salin berikut ini ke file bernama `role1.json`. Ganti `your-aws-account-id` dengan AWSID akun dan `aws-region` dengan AWS Wilayah Anda AWS sumber daya.

Note

Contoh ini mencakup kunci konteks kondisi global untuk melindungi terhadap masalah keamanan wakil yang bingung. Untuk informasi selengkapnya, lihat [the section called "Pencegahan wakil bingung lintas layanan"](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "sagemaker.amazonaws.com",
          "iotanalytics.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "your-aws-account-id"
        }
      }
    }
  ]
}

```

```

        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:iotanalytics:aws-region:your-aws-account-id:dataset/DOC-EXAMPLE-DATASET"
        }
    }
]
}

```

2. Buat peran yang memberikan izin akses SageMaker dan AWS IoT Analytics, menggunakan file `role1.json` yang Anda unduh.

```
aws iam create-role --role-name container-execution-role --assume-role-policy-document file://role1.json
```

3. Unduh berikut ini ke file bernama `policy1.json` dan gantilah *your-account-id* dengan ID akun Anda (lihat ARN kedua di bawah `Statement:Resource`).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:PutObject",
        "s3:GetObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::*-dataset-*/*",
        "arn:aws:s3:::octave-sample-data-your-account-id/*"
      ],
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [

```

```

    "ecr:GetAuthorizationToken",
    "ecr:GetDownloadUrlForLayer",
    "ecr:BatchGetImage",
    "ecr:BatchCheckLayerAvailability",
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:DescribeLogStreams",
    "logs:GetLogEvents",
    "logs:PutLogEvents"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:ListAllMyBuckets"
  ],
  "Resource": "*"
}
]
}

```

4. Buat kebijakan IAM, menggunakan `policy.json` yang baru saja Anda unduh.

```
aws iam create-policy --policy-name ContainerExecutionPolicy --policy-document
file://policy1.json
```

5. Melampirkan kebijakan pada peran tersebut.

```
aws iam attach-role-policy --role-name container-execution-role --policy-arn
arn:aws:iam::your-account-id:policy/ContainerExecutionPolicy
```

Langkah 5: Membuat dataset dengan tindakan kontainer

1. Unduh yang berikut ini ke file bernama `cli-input.json` dan mengganti semua contoh `your-account-id` dan `region` dengan nilai yang sesuai.

```

{
  "datasetName": "octave_dataset",
  "actions": [

```

```

    {
      "actionName": "octave",
      "containerAction": {
        "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-
moment",
        "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
        "resourceConfiguration": {
          "computeType": "ACU_1",
          "volumeSizeInGB": 1
        },
        "variables": [
          {
            "name": "octaveResultS3URI",
            "outputFileUriValue": {
              "fileName": "output.mat"
            }
          },
          {
            "name": "inputDataS3BucketName",
            "stringValue": "octave-sample-data-your-account-id"
          },
          {
            "name": "inputDataS3Key",
            "stringValue": "input.txt"
          },
          {
            "name": "order",
            "stringValue": "3"
          }
        ]
      }
    }
  ]
}

```

2. Buat set data menggunakan `filecli-input.json` Anda baru saja diunduh dan diedit.

```
aws iotanalytics create-dataset --cli-input-json file://cli-input.json
```

Langkah 6: Memanggil pembuatan konten set data

1. Jalankan perintah berikut.

```
aws iotanalytics create-dataset-content --dataset-name octave-dataset
```

Langkah 7: Dapatkan konten set data

1. Jalankan perintah berikut.

```
aws iotanalytics get-dataset-content --dataset-name octave-dataset --version-id \
$LATEST
```

2. Anda mungkin perlu menunggu beberapa menit sampai DatasetContentState adalah SUCCEEDED.

Langkah 8: Cetak output pada Octave

1. Gunakan shell Octave untuk mencetak output dari wadah dengan menjalankan perintah berikut.

```
bash> octave
octave> load output.mat
octave> disp(M)
-0.016393 -0.098061 0.380311 -0.564377 -1.318744
```


Memvisualisasi AWS IoT Analytics data

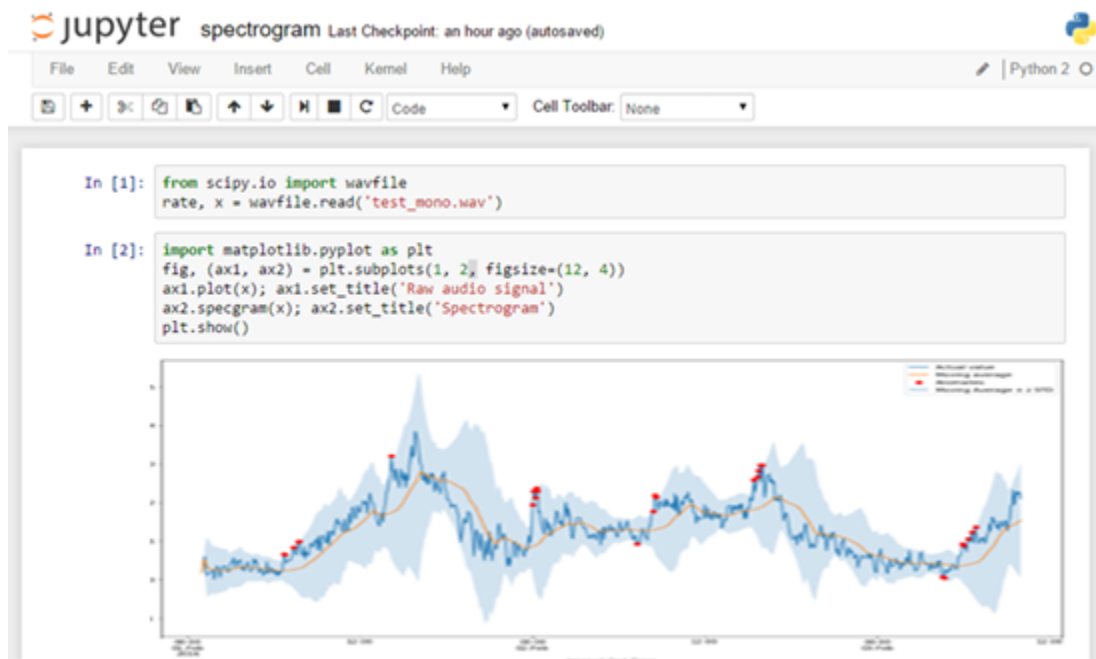
Untuk memvisualisasikan AWS IoT Analytics Data, Anda dapat menggunakan AWS IoT Analytics konsol atau Amazon QuickSight.

Topik

- [Memvisualisasi AWS IoT Analytics data dengan konsol](#)
- [Memvisualisasi AWS IoT Analytics Data dengan Amazon QuickSight](#)

Memvisualisasi AWS IoT Analytics data dengan konsol

AWS IoT Analytics dapat menanamkan output HTML dari dataset kontainer Anda (ditemukan dalam file output .html) pada halaman konten dataset kontainer [AWS IoT Analytics konsol](#). Misalnya, jika Anda mendefinisikan dataset kontainer yang menjalankan notebook Jupyter, dan Anda membuat visualisasi di notebook Jupyter Anda, dataset Anda mungkin terlihat seperti berikut.



Kemudian, setelah konten dataset kontainer dibuat, Anda dapat melihat visualisasi ini pada konsol Set Data halaman konten.



Untuk informasi tentang membuat dataset kontainer yang menjalankan notebook Jupyter, lihat [Mengotomatisasi alur kerja Anda](#).

Memvisualisasi AWS IoT Analytics Data dengan Amazon QuickSight

AWS IoT Analytics menyediakan integrasi langsung dengan [Amazon QuickSight](#). Amazon QuickSight adalah layanan analitik bisnis yang cepat yang dapat Anda gunakan untuk membangun visualisasi, melakukan analisis ad-hoc, dan mendapatkan wawasan bisnis dari data Anda dengan cepat. Amazon QuickSight memungkinkan organisasi untuk menskalakan ratusan ribu pengguna, dan memberikan kinerja responsif dengan menggunakan mesin dalam memori (SPICE) yang kuat. Anda dapat memilih AWS IoT Analytics dataset di Amazon QuickSight konsol dan mulai membuat dashboard dan visualisasi. Amazon QuickSight tersedia di [Wilayah ini](#).

Untuk memulai dengan Amazon Anda QuickSight visualisasi, Anda harus membuat Amazon QuickSight akun. Pastikan Anda memberikan Amazon QuickSight akses ke AWS IoT Analytics data saat Anda mengatur akun Anda. Jika Anda sudah memiliki akun, berikan Amazon QuickSight akses AWS IoT Analytics data dengan memilih Admin, Kelola QuickSight, Keamanan & izin. Di bawah Akses QuickSight ke AWS jasa, pilih Menambah atau menghapus, lalu pilih kotak centang di samping AWS IoT Analytics dan pilih Perbarui.

QuickSight

Account name: [redacted]
Edition: Enterprise

Manage users
Your subscriptions
SPICE capacity
Account settings
Security & permissions
Manage VPC connections
Domains and Embedding

Security & permissions

QuickSight can control access to AWS resources for the entire account in addition to individual users and groups

QuickSight access to AWS services

Amazon Redshift Amazon RDS IAM Amazon S3 AWS IoT Analytics

By configuring access to AWS services, QuickSight can access the data in those services. Access by users and groups can be controlled through the options below.

[Add or remove](#)

Default resource access

① Users and groups have access to all connected resources.

QuickSight can allow or deny access to all users and groups by default, when an individual access control is not in effect for a particular user or group

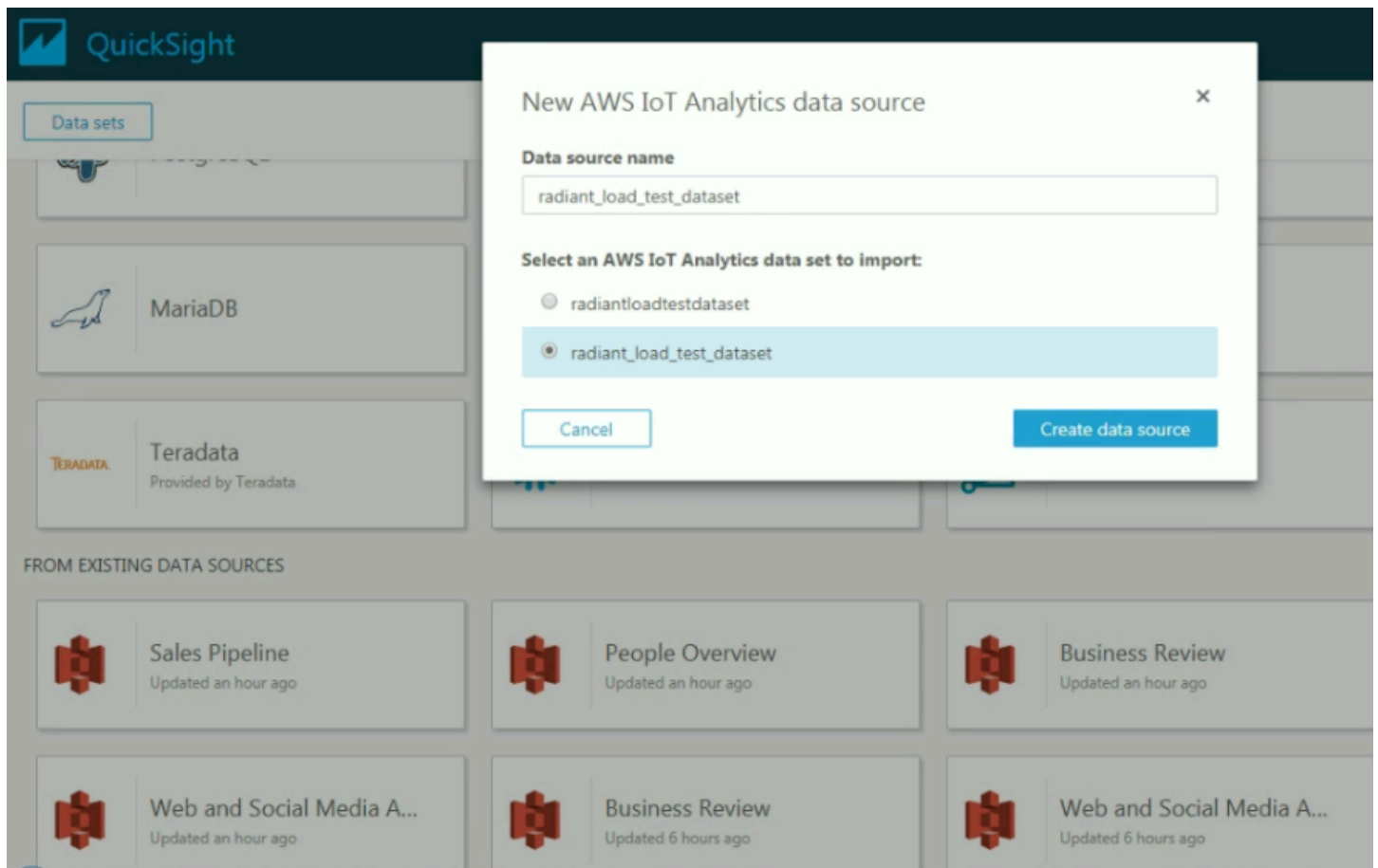
[Change](#)

Resource access for individual users and groups

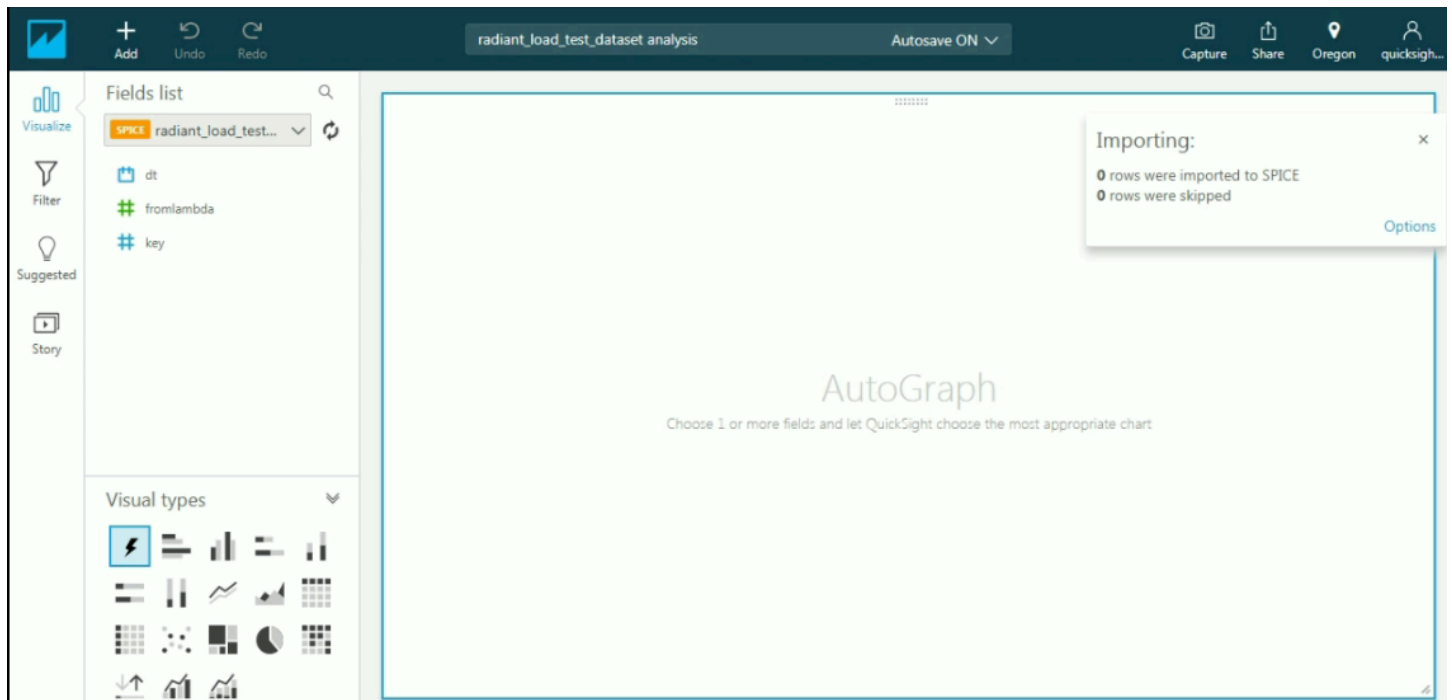
Resource access is controlled by assigning IAM policies.

[IAM policy assignments](#)

Setelah akun Anda diatur, dari admin Amazon QuickSight laman konsol pilih Analisis Baru dan Set data baru, dan kemudian pilih AWS IoT Analytics sebagai sumber. Masukkan nama untuk sumber data Anda, pilih dataset yang akan diimpor, lalu pilih Membuat sumber data.



Setelah sumber data dibuat, Anda dapat membuat visualisasi di Amazon QuickSight.



Untuk informasi tentang Amazon QuickSight dashboard dan dataset, lihat [Amazon QuickSight dokumentasi](#).

Menandai Sumber Daya AWS IoT Analytics Anda

Untuk membantu Anda mengelola saluran, kumpulan data, penyimpanan data, dan saluran, Anda dapat menetapkan metadata Anda sendiri ke setiap sumber daya tersebut dalam bentuk tanda. Bab ini menjelaskan tag dan menunjukkan kepada Anda cara menciptakannya.

Topik

- [Dasar tanda](#)
- [Menggunakan tanda dengan kebijakan IAM](#)
- [Pembatasan tanda](#)

Dasar tanda

Tag mengizinkan Anda untuk mengategorikan sumber daya AWS IoT Analytics Anda dengan berbagai cara, misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Hal ini berguna ketika Anda memiliki banyak sumber daya dengan jenis yang sama—Anda dapat dengan cepat mengidentifikasi sumber daya tertentu berdasarkan tanda yang telah Anda tetapkan. Setiap tanda terdiri dari kunci dan nilai opsional, yang keduanya Anda tentukan. Misalnya, Anda dapat menentukan serangkaian tanda untuk saluran Anda yang dapat membantu Anda melacak jenis perangkat yang bertanggung jawab atas setiap sumber pesan. Sebaiknya Anda merancang seperangkat kunci tanda yang memenuhi kebutuhan Anda untuk setiap jenis sumber daya. Penggunaan seperangkat kunci tanda yang konsisten akan mempermudah Anda dalam mengelola sumber daya Anda. Anda dapat mencari dan menyaring sumber daya berdasarkan tanda yang Anda tambahkan.

Anda juga dapat menggunakan tanda untuk mengategorikan dan melacak biaya. Saat Anda menerapkan tanda untuk saluran, kumpulan data, penyimpanan data, atau saluran, AWS membuat laporan alokasi biaya sebagai file CSV (comma separated value) dengan penggunaan dan biaya yang diagregasi oleh tanda Anda. Anda dapat menerapkan tanda yang mewakili kategori bisnis (seperti pusat biaya, nama aplikasi, atau pemilik) untuk mengatur biaya Anda di berbagai layanan. Untuk informasi selengkapnya tentang penggunaan tanda untuk alokasi biaya, lihat [Gunakan Tanda Alokasi Biaya](#) dalam [Panduan AWS Billing Pengguna](#).

Untuk kemudahan penggunaan, gunakan Tag Editor di AWS Billing and Cost Management konsol, yang menyediakan cara terpusat dan terpadu untuk membuat dan mengelola tanda Anda. Untuk informasi selengkapnya, lihat [Bekerja dengan Editor Tag](#) di [Memulai dengan AWS Management Console](#).

Anda juga dapat bekerja dengan tag menggunakan AWS CLI dan AWS IoT Analytics API. Anda dapat mengaitkan tag dengan saluran, kumpulan data, penyimpanan data, dan jaringan pipa saat Anda membuatnya; gunakan bidang Tag dalam perintah berikut:

- [CreateChannel](#)
- [CreateDataset](#)
- [CreateDatastore](#)
- [CreatePipeline](#)

Anda dapat menambahkan, memodifikasi, atau menghapus tanda untuk sumber daya yang ada yang mendukung penandaan. Gunakan perintah berikut:

- [TagResource](#)
- [ListTagsForResource](#)
- [UntagResource](#)

Anda dapat mengedit kunci dan nilai tanda, dan Anda dapat membuang tanda dari sumber daya kapan saja. Anda dapat mengatur nilai tanda ke string kosong, akan tetapi Anda tidak dapat mengatur nilai tanda ke nol. Jika Anda menambahkan tag yang memiliki kunci yang sama dengan tag yang ada pada sumber daya tersebut, nilai yang baru akan menimpa nilai olde. Jika Anda menghapus sebuah sumber daya, semua tanda yang terkait dengan sumber daya tersebut juga dihapus.

Menggunakan tanda dengan kebijakan IAM

Anda dapat menggunakan `Condition` elemen (juga disebut `Condition` blok) dengan kunci konteks syarat berikut dalam kebijakan IAM untuk mengontrol akses pengguna (izin) berdasarkan tanda sumber daya:

- Gunakan `iotanalytics:ResourceTag/<tag-key>: <tag-value>` yo mengizinkan atau tolak tindakan pengguna pada sumber daya dengan tanda tertentu.
- Gunakan `aws:RequestTag/<tag-key>: <tag-value>` untuk mengharuskan penggunaan (atau tidak digunakan) saat membuat permintaan API agar membuat atau memodifikasi sumber daya yang mengizinkan tanda.

- Gunakan `aws:TagKeys: [<tag-key>, ...]` untuk mengharuskan penggunaan (atau tidak digunakan) saat membuat permintaan API agar membuat atau memodifikasi sumber daya yang mengizinkan tanda.

Note

Kunci/nilai konteks syarat dalam kebijakan IAM hanya berlaku untuk AWS IoT Analytics tindakan tersebut di mana pengidentifikasi untuk sumber daya yang dapat ditandai adalah parameter yang diperlukan. Misalnya, penggunaan [DescribeLoggingOptions](#) tidak diperbolehkan/ditolak berdasarkan kunci/nilai konteks syarat, karena tidak ada sumber daya yang dapat ditandai (saluran, kumpulan data, penyimpanan data atau pipa) direferensikan dalam permintaan ini.

Untuk informasi selengkapnya, lihat [Mengontrol akses menggunakan tag](#) di Panduan Pengguna IAM. Bagian [Referensi kebijakan JSON IAM](#) dari panduan tersebut menyajikan sintaks, deskripsi, dan contoh terperinci elemen, variabel, dan logika evaluasi kebijakan JSON dalam IAM.

Kebijakan contoh berikut memberlakukan pembatasan berbasis dua. Pengguna yang dibatasi oleh kebijakan ini:

1. Tidak dapat memberikan sumber daya tag "env=prod" (lihat baris "`aws:RequestTag/env`" : "prod" dalam contoh).
2. Tidak dapat memodifikasi atau mengakses sumber daya yang ada tanda "env=prod" (lihat baris "`iotanalytics:ResourceTag/env`" : "prod" dalam contoh).

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "aws:RequestTag/env" : "prod"
        }
      }
    }
  ]
}
```



```

    },
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "iotanalytics:ResourceTag/env" : "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    }
  ]
}

```

Anda juga dapat menentukan beberapa nilai tanda dengan memasukkannya ke dalam daftar, seperti contoh berikut.

```

"StringEquals" : {
  "iotanalytics:ResourceTag/env" : ["dev", "test"]
}

```

Note

Jika Anda mengizinkan/menolak akses pengguna ke sumber daya berdasarkan tanda, penting untuk mempertimbangkan untuk secara eksplisit menolak memberikan kemampuan kepada pengguna untuk menambahkan atau membuangnya dari sumber daya yang sama. Jika tidak, pengguna dapat mengakali pembatasan Anda dan mendapatkan akses atas sumber daya dengan memodifikasi tandanya.

Pembatasan tanda

Batasan dasar berikut berlaku untuk tag:

- Jumlah maksimum tanda — 50
- Panjang kunci maksimum — 127 karakter dalam UTF-8
- Panjang nilai maksimum — 255 karakter dalam UTF-8
- Kunci dan nilai tanda peka huruf besar dan kecil.
- Jangan menggunakan namaaws: prefix atau nilai tag Anda, karena ini khusus untukAWS digunakan. Anda tidak dapat mengedit atau menghapus nama atau nilai tanda dengan prefiks ini. Tanda dengan awalan ini tidak memengaruhi tanda Anda per batas sumber.
- Jika skema penandaan Anda digunakan di beberapa layanan dan sumber daya, ingatlah bahwa layanan lain mungkin memiliki pembatasan pada karakter yang diizinkan. Umumnya, karakter yang diperbolehkan adalah: huruf, spasi, dan angka yang dapat mewakili dalam UTF-8, serta karakter berikut: + - =. _:/@.

Ekspresi SQL diAWS IoT Analytics

Set data yang dihasilkan menggunakan ekspresi SQL pada data di penyimpanan data.AWS IoT Analyticsmenggunakan kueri SQL, fungsi dan operator yang sama seperti Amazon Athena.

AWS IoT Analyticsmendukung subset ANSI sintaks SQL standar.

```
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[[ INNER | OUTER ] LEFT | RIGHT | FULL | CROSS JOIN join_item [ ON join_condition ]]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ UNION [ ALL | DISTINCT ] union_query ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ LIMIT [ count | ALL ] ]
```

Untuk deskripsi parameter, lihat[Parameter](#)diDokumentasi Amazon Athena.

AWS IoT Analyticsdan Amazon Athena tidak mendukung hal berikut:

- WITHklausa
- CREATE TABLE AS SELECTpernyataan
- INSERT INTOpernyataan
- Pernyataan siap, Anda tidak dapat menjalankanEXECUTEbersamaUSING.
- CREATE TABLE LIKE
- DESCRIBE INPUT dan DESCRIBE OUTPUT
- EXPLAINpernyataan
- Fungsi yang ditentukan pengguna (UDF atau UDAF)
- Prosedur tersimpan
- Konektor gabungan

Topik

- [Fungsionalitas SQL yang didukung diAWS IoT Analytics](#)
- [Memecahkan masalah umum dengan query SQL diAWS IoT Analytics](#)

Fungsionalitas SQL yang didukung diAWS IoT Analytics

Dataset yang dihasilkan dengan menggunakan ekspresi SQL pada data di sebuah penyimpanan data. Query Anda menjalankan diAWS IoT Analytics didasarkan pada [Presto 0.217](#).

Tipe data yang didukung

AWS IoT Analytics dan Amazon Athena mendukung tipe data ini.

- primitive_type
 - TINYINT
 - SMALLINT
 - INT
 - BIGINT
 - BOOLEAN
 - DOUBLE
 - FLOAT
 - STRING
 - TIMESTAMP
 - DECIMAL(precision, scale)
 - DATE
 - CHAR(data karakter panjang tetap dengan panjang tertentu)
 - VARCHAR(Data karakter variabel-panjang dengan panjang tertentu)
- array_type
 - ARRAY<data_type>
- map_type
 - MAP<primitive_type, data_type>
- struct_type
 - STRUCT<col_name:data_type[COMMENT col_comment][,...]>

Note

AWS IoT Analytics dan Amazon Athena tidak mendukung beberapa tipe data.

Fungsi yang didukung

Fungsionalitas Amazon Athena dan AWS IoT Analytics SQL didasarkan pada [Presto 0.217](#). Untuk informasi tentang fungsi, operator, dan ekspresi terkait, lihat [Fungsi dan Operator](#) dan bagian tertentu berikut dari dokumentasi Presto.

- Operator logistik
- Fungsi dan operator perbandingan
- Ekspresi Bersyarat
- Fungsi konversi
- Fungsi dan operator matematika
- Fungsi bitwise
- Fungsi desimal dan operator
- fungsi String dan operator
- Fungsi biner
- Fungsi tanggal dan waktu
- Fungsi ekspresi reguler
- fungsi JSON dan operator
- fungsi URL
- Fungsi agregat
- Fungsi jendela
- Fungsi warna
- fungsi array dan operator
- Fungsi peta dan operator
- Ekspresi Lambda
- Fungsi teradata

Note

AWS IoT Analytics dan Amazon Athena tidak mendukung fungsi yang ditentukan pengguna (UDF atau UDAF) atau prosedur yang disimpan.

Memecahkan masalah umum dengan query SQL di AWS IoT Analytics

Informasi berikut dapat membantu Anda memecahkan masalah dengan kueri SQL AWS IoT Analytics.

- Untuk melarikan diri dari satu kutipan, mendahului dengan kutipan tunggal lain. Jangan bingung dengan kutipan ganda.

Example Contoh

```
SELECT '0' 'Reilly'
```

- Untuk melarikan diri garis bawah, gunakan backticks untuk melampirkan nama kolom penyimpanan data yang dimulai dengan garis bawah.

Example Contoh

```
SELECT `_myMessageAttribute` FROM myDataStore
```

- Untuk melarikan diri nama dengan angka, lampirkan nama penyimpanan data yang mencakup angka dalam tanda kutip ganda.

Example Contoh

```
SELECT * FROM "myDataStore123"
```

- Untuk melarikan diri kata kunci reserved, lampirkan kata kunci reserved dalam tanda kutip ganda. Untuk informasi selengkapnya, lihat [Daftar Kata Kunci Cadangan](#) di Pernyataan SQL SELECT.

Keamanan di AWS IoT Analytics

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](#) menggambarkan ini sebagai keamanan cloud dan keamanan di 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. Efektivitas keamanan kami diuji dan diverifikasi secara rutin oleh auditor pihak ketiga sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku AWS IoT Analytics, lihat [AWS layanan dalam cakupan berdasarkan program kepatuhan](#).
- Keamanan di cloud - Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data, persyaratan perusahaan, serta hukum dan peraturan yang berlaku.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS IoT Analytics. Topik berikut menunjukkan cara mengonfigurasi AWS IoT Analytics untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga akan belajar cara menggunakan AWS layanan lain yang dapat membantu Anda memantau dan mengamankan AWS IoT Analytics sumber daya Anda.

AWS Identity and Access Management di AWS IoT Analytics

AWS Identity and Access Management (IAM) adalah AWS layanan 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 Analytics IAM adalah AWS layanan yang dapat Anda gunakan tanpa biaya tambahan.

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS IoT Analytics

Pengguna layanan — Jika Anda menggunakan AWS IoT Analytics layanan untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS IoT Analytics 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 Analytics, lihat [Memecahkan masalah AWS IoT Analytics identitas dan akses](#).

Administrator layanan — Jika Anda bertanggung jawab atas AWS IoT Analytics sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS IoT Analytics. Tugas Anda adalah menentukan AWS IoT Analytics fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS IoT Analytics, lihat [Bagaimana AWS IoT Analytics 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 Analytics. Untuk melihat contoh kebijakan AWS IoT Analytics berbasis identitas yang dapat Anda gunakan di IAM, lihat [AWS IoT Analytics contoh kebijakan berbasis identitas](#)

Mengautentikasi dengan identitas

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 kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas gabungan, 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. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di 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 lebih lanjut, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) di AWS](#) 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 Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar tugas lengkap 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, sebaiknya andalkan kredensial temporer, dan bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, sebaiknya rotasikan 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 kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama 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 tersebut dimaksudkan untuk dapat diambil oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, silakan lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) 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. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi diautentikasi, identitas tersebut dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi sekumpulan izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengaitkan izin yang ditetapkan ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat [Rangkaian 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 (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Contoh, ketika Anda melakukan panggilan dalam layanan, umumnya layanan tersebut menjalankan aplikasi

di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, 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 tindakan yang kemudian memulai 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 [Meneruskan sesi akses](#).
- Peran IAM – Peran layanan adalah [peran IAM](#) yang diambil layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat 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 mengambil peran untuk melakukan sebuah 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 kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

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 [Ikhtisar kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja 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 pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. 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 terkelola atau kebijakan inline, lihat memilih [antara kebijakan terkelola dan kebijakan inline](#) di Panduan Pengguna IAM.

Jenis kebijakan lainnya

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- **Batasan izin** – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna 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 secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- **Kebijakan kontrol layanan (SCP)** — SCP 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 organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke sebagian atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .
- **Kebijakan sesi** – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. 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 di salah satu kebijakan ini akan membatalkan izin tersebut. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Jika beberapa jenis kebijakan diberlakukan untuk satu 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.

Bagaimana AWS IoT Analytics bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses AWS IoT Analytics, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan. AWS IoT Analytics Untuk mendapatkan pandangan tingkat tinggi tentang bagaimana AWS IoT Analytics dan AWS layanan lain bekerja dengan IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Topik di halaman ini:

- [AWS IoT Analytics kebijakan berbasis identitas](#)
- [AWS IoT Analytics kebijakan berbasis sumber daya](#)
- [Otorisasi berdasarkan tag AWS IoT Analytics](#)
- [AWS IoT Analytics Peran IAM](#)

AWS IoT Analytics kebijakan berbasis identitas

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. AWS IoT Analytics 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

Elemen `Action` kebijakan berbasis identitas IAM menjelaskan tindakan atau tindakan tertentu yang akan diizinkan atau ditolak oleh kebijakan tersebut. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Tindakan tersebut digunakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan AWS IoT Analytics menggunakan awalan berikut sebelum tindakan:

Misalnya, `iotanalytics:` untuk memberikan izin kepada seseorang untuk membuat AWS IoT Analytics channel dengan operasi AWS IoT Analytics `CreateChannel` API, Anda menyertakan `iotanalytics:BatchPuMessage` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus mencakup salah satu `Action` atau `NotAction` elemen. AWS IoT Analytics 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": [  
  "iotanalytics:action1",  
  "iotanalytics:action2"  
]
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut.

```
"Action": "iotanalytics:Describe*"
```

Untuk melihat daftar AWS IoT Analytics tindakan, lihat [Tindakan yang ditentukan oleh AWS IoT Analytics](#) dalam Panduan Pengguna IAM.

Sumber daya

Elemen `Resource` menentukan objek di mana tindakan berlaku. Pernyataan harus mencakup elemen `Resource` atau `NotResource`. Anda menentukan sumber daya menggunakan ARN atau menggunakan wildcard (*) untuk menunjukkan bahwa pernyataan berlaku untuk semua sumber daya.

Sumber daya AWS IoT Analytics dataset memiliki ARN berikut.

```
arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/${DatasetName}
```

Untuk informasi selengkapnya tentang format ARN, lihat [Amazon Resource Name \(ARN\) dan ruang nama layanan AWS](#).

Misalnya, untuk menentukan Foobar kumpulan data dalam pernyataan Anda, gunakan ARN berikut.

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/Foobar"
```

Untuk menentukan semua instance milik akun tertentu, gunakan wildcard (*).

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/*"
```

Beberapa AWS IoT Analytics tindakan, seperti untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

```
"Resource": "*" 
```


Beberapa tindakan AWS IoT Analytics API melibatkan banyak sumber daya. Misalnya, `CreatePipeline` referensi sebagai saluran dan kumpulan data, sehingga pengguna harus memiliki izin untuk menggunakan saluran dan kumpulan data. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma.

```
"Resource": [  
  "resource1",  
  "resource2"  
]
```

Untuk melihat daftar jenis AWS IoT Analytics sumber daya dan ARNnya, lihat [Sumber daya yang ditentukan oleh AWS IoT Analytics](#) dalam Panduan Pengguna IAM. Untuk mempelajari tindakan yang dapat menentukan ARN setiap sumber daya, lihat [Tindakan yang ditentukan AWS IoT Analytics](#).

Kunci syarat

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan ketentuan yang mengizinkan Anda untuk menerapkan pernyataan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), seperti sama dengan atau kurang dari, untuk mencocokkan ketentuan dalam kebijakan dengan nilai dalam permintaan.

Jika Anda menentukan beberapa elemen `Condition` dalam pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS mengevaluasinya menggunakan operasi AND. Jika Anda menentukan beberapa nilai untuk satu kunci persyaratan, AWS akan mengevaluasi syarat tersebut menggunakan operasi OR yang logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama pengguna mereka. Untuk informasi lebih lanjut, lihat [Elemen kebijakan IAM: Variabel dan tanda](#) dalam Panduan Pengguna IAM.

AWS IoT Analytics tidak menyediakan kunci kondisi khusus tujuh, tetapi mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) dalam Panduan Pengguna IAM.

Contoh-contoh

Untuk melihat contoh kebijakan AWS IoT Analytics berbasis identitas, lihat [AWS IoT Analytics contoh kebijakan berbasis identitas](#)

AWS IoT Analytics kebijakan berbasis sumber daya

AWS IoT Analytics tidak mendukung kebijakan berbasis sumber daya. Untuk melihat contoh halaman kebijakan berbasis sumber daya terperinci, lihat [Menggunakan kebijakan berbasis sumber daya di Panduan Pengembang](#). AWS Lambda AWS Lambda

Otorisasi berdasarkan tag AWS IoT Analytics

Anda dapat melampirkan tag ke AWS IoT Analytics sumber daya atau meneruskan tag dalam permintaan AWS IoT Analytics. Untuk mengontrol akses berdasarkan tag, Anda memberikan informasi tag dalam [elemen kondisi](#) kebijakan menggunakan `iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}` atau kunci `aws:TagKeys` kondisi. Untuk informasi selengkapnya tentang menandai AWS IoT Analytics sumber daya, lihat [Menandai sumber daya Anda AWS IoT Analytics](#).

Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke sumber daya berdasarkan tag pada sumber daya tersebut, lihat [Melihat AWS IoT Analytics saluran berdasarkan tag](#).

AWS IoT Analytics Peran IAM

[IAM role](#) adalah entitas dalam Akun AWS Anda yang memiliki izin khusus.

Menggunakan kredensi sementara dengan AWS IoT Analytics

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensial keamanan sementara dengan memanggil AWS Security Token Service (AWS STS) operasi API seperti [AssumeRole](#) atau [GetFederationToken](#).

AWS IoT Analytics tidak mendukung penggunaan kredensial sementara.

Peran terkait layanan

[Peran berlapis AWS layanan](#) memungkinkan layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran terkait layanan muncul di akun IAM Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat tetapi tidak dapat mengedit izin untuk peran terkait layanan.

AWS IoT Analytics tidak mendukung peran terkait layanan.

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 akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

AWS IoT Analytics mendukung peran layanan.

Pencegahan wakil bingung lintas layanan

Masalah wakil yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Masuk AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil bingung. Peniruan lintas layanan dapat terjadi ketika satu layanan (layanan panggilan) panggilan layanan lain (yang disebut layanan). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak pada 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 Anda untuk semua layanan, dengan prinsipal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan [aws:SourceArn](#) dan [aws:SourceAccount](#) kunci konteks kondisi global dalam kebijakan sumber daya. Ini membatasi izin yang AWS IoT Analytics memberikan layanan lain ke sumber daya. Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun di `aws:SourceArn` nilai harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan kebijakan yang sama.

Cara paling efektif untuk melindungi terhadap masalah wakil yang bingung adalah dengan menggunakan `aws:SourceArn` kunci konteks kondisi global dengan Amazon Resource Name (ARN) sumber daya yang lengkap. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan `aws:SourceArn` kunci konteks kondisi global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Sebagai contoh, `arn:aws:iotanalytics::123456789012:*`.

Topik

- [Pencegahan untuk Amazon S3 ember](#)
- [Pencegahan dengan Amazon CloudWatch Beberapa catatan](#)
- [Bingung wakil pencegahan untuk pelanggan dikelola AWS IoT Analytics sumber daya](#)

Pencegahan untuk Amazon S3 ember

Jika Anda menggunakan penyimpanan Amazon S3 yang dikelola pelanggan untuk Anda AWS IoT Analytics penyimpanan data, bucket Amazon S3 yang menyimpan data Anda mungkin terpapar masalah wakil yang membingungkan.

Misalnya, Nikki Wolf menggunakan ember Amazon S3 milik pelanggan yang disebut *DOC-CONTOH-EMBER*. Bucket menyimpan informasi untuk AWS IoT Analytics penyimpanan data yang dibuat di Wilayah *us-east-1*. Dia menetapkan kebijakan yang memungkinkan AWS IoT Analytics Layanan utama untuk query *DOC-CONTOH-EMBER* atas namanya. Rekan kerja Nikki, Li Juan, pertanyaan *DOC-CONTOH-EMBER* dari akunnya sendiri dan membuat dataset dengan hasilnya. Hasilnya, AWS IoT Analytics kepala layanan menanyakan ember Amazon S3 Nikki atas nama Li meskipun Li menjalankan kueri dari akunnya.

Untuk mencegah hal ini, Nikki dapat menentukan `aws:SourceAccount` kondisi atau `aws:SourceArn` kondisi dalam kebijakan untuk *DOC-CONTOH-EMBER*.

Menentukan `aws:SourceAccount` kondisi- Contoh kebijakan bucket berikut menetapkan bahwa hanya AWS IoT Analytics sumber daya dari akun Nikki (*123456789012*) dapat mengakses *DOC-CONTOH-EMBER*.

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
```

```

        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "123456789012"
        }
    }
}
]
}

```

Menentukan **aws:SourceArn** kondisi- Atau, Nikki dapat menggunakan **aws:SourceArn** kondisi.

```

{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-EXAMPLE-DATASET",

```

```

    "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
    ]
  }
}

```

Pencegahan dengan Amazon CloudWatch Beberapa catatan

Anda dapat mencegah masalah confused deputy saat memantau dengan Amazon CloudWatch Log. Kebijakan sumber daya berikut menunjukkan cara mencegah masalah wakil yang bingung dengan:

- Kunci konteks kondisi global, `aws:SourceArn`
- `Klasteraws:SourceAccount` dengan AWSID akun
- Sumber daya pelanggan yang berkaitan dengan `sts:AssumeRole` request AWS IoT Analytics

Ganti `123456789012` dengan AWSID akun, dan `us-east-1` dengan Wilayah AWS IoT Analytics akun dalam contoh berikut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "logs:PutLogEvents",
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:us-east-1:123456789012:*/*"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}

```

Untuk informasi selengkapnya tentang mengaktifkan dan mengonfigurasi Amazon CloudWatch Log, lihat [the section called “Pencatatan dan pemantauan”](#).

Bingung wakil pencegahan untuk pelanggan dikelola AWS IoT Analytics sumber daya

Jika Anda memberikan AWS IoT Analytics izin untuk kemudian melakukan tindakan pada AWS IoT Analytics sumber daya, sumber daya dapat terkena masalah wakil bingung. Untuk mencegah masalah wakil yang bingung, Anda dapat membatasi izin yang diberikan kepada AWS IoT Analytics dengan kebijakan sumber daya contoh berikut.

Topik

- [Pencegahan untuk AWS IoT Analytics penyimpanan saluran dan data](#)
- [Pencegahan wakil bingung lintas layanan untuk AWS IoT Analytics saluran pengiriman konten set data](#)

Pencegahan untuk AWS IoT Analytics penyimpanan saluran dan data

Anda menggunakan IAM role untuk mengontrol AWS sumber daya yang AWS IoT Analytics dapat mengakses atas nama Anda. Untuk mencegah mengekspos peran Anda pada masalah wakil yang bingung, Anda dapat menentukan AWS akun di `aws:SourceAccount` elemen dan ARN dari AWS IoT Analytics sumber daya di `aws:SourceArn` elemen kebijakan kepercayaan yang Anda lampirkan pada peran.

Pada contoh berikut, ganti `123456789012` dengan AWS ID akun dan `arn:aws:iotanalytics:aws-region:123456789012:Saluran/D0C-Contoh-saluran` dengan ARN dari AWS IoT Analytics saluran atau penyimpanan data.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

```

    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL"
    }
  }
}
]
}

```

Untuk mempelajari lebih lanjut tentang opsi penyimpanan S3 yang dikelola pelanggan untuk saluran dan penyimpanan data, lihat [CustomerManagedChannelS3Storage](#) dan [CustomerManagedDatastoreS3Storage](#) di dalam [AWS IoT Analytics Referensi API](#).

Pencegahan wakil bingung lintas layanan untuk AWS IoT Analytics saluran pengiriman konten set data

Peran IAM role IAM role AWS IoT Analytics mengasumsikan untuk mengirimkan hasil kueri set data ke Amazon S3 atau ke AWS IoT Events dapat terkena masalah wakil yang membingungkan. Untuk mencegah masalah confused deputy, tentukan `aws:SourceAccount` elemen dan ARN dari AWS IoT Analytics sumber daya di `aws:SourceArn` elemen kebijakan kepercayaan yang Anda lampirkan pada peran Anda.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExampleTrustPolicyDocument",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:dataset/DOC-EXAMPLE-DATASET"
        }
      }
    }
  ]
}

```

```
}  
]  
}
```

Untuk detail selengkapnya tentang mengonfigurasi aturan pengiriman konten set data, lihat [contentDeliveryRules](#) di dalam [AWS IoT Analytics Referensi API](#).

AWS IoT Analytics contoh kebijakan berbasis identitas

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi AWS IoT Analytics sumber daya. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan 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 di Panduan Pengguna IAM](#)

Topik di halaman ini:

- [Praktik terbaik kebijakan](#)
- [Menggunakan AWS IoT Analytics konsol](#)
- [Izinkan para pengguna untuk melihat izin mereka sendiri](#)
- [Mengakses satu masukan AWS IoT Analytics](#)
- [Melihat AWS IoT Analytics saluran berdasarkan tag](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas adalah pilihan yang sangat tepat. Mereka menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS IoT Analytics sumber daya di akun Anda. Tindakan ini dapat menimbulkan biaya untuk akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulai menggunakan kebijakan AWS terkelola - Untuk mulai menggunakan AWS IoT Analytics dengan cepat, gunakan kebijakan AWS terkelola untuk memberi karyawan Anda izin yang mereka butuhkan. Kebijakan ini sudah tersedia di akun Anda dan dikelola serta diperbarui oleh AWS. Untuk informasi selengkapnya, lihat [Memulai menggunakan izin dengan kebijakan AWS terkelola](#) di Panduan Pengguna IAM.

- Berikan hak istimewa paling sedikit - Saat Anda membuat kebijakan khusus, berikan hanya izin yang diperlukan untuk melakukan tugas. Mulai dengan satu set izin minimum dan berikan izin tambahan sesuai kebutuhan. Melakukan hal tersebut lebih aman daripada memulai dengan izin yang terlalu fleksibel, lalu mencoba memperketatnya nanti. Untuk informasi selengkapnya, lihat [Pemberian hak istimewa terendah](#) dalam Panduan Pengguna IAM.
- Aktifkan MFA untuk operasi sensitif - Untuk keamanan ekstra, pengguna harus menggunakan otentikasi multi-faktor (MFA) untuk mengakses sumber daya sensitif atau operasi API. Untuk informasi selengkapnya, lihat [Menggunakan autentikasi multifaktor \(MFA\) dalam AWS](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi kebijakan untuk keamanan ekstra - Sejauh praktis, tentukan kondisi di mana kebijakan berbasis identitas Anda mengizinkan akses ke sumber daya. Misalnya, Anda dapat menulis kondisi untuk menentukan rentang alamat IP yang diijinkan yang harus berasal dari permintaan. Anda juga dapat menulis persyaratan untuk mengizinkan permintaan hanya dalam rentang tanggal atau waktu tertentu, atau untuk mewajibkan penggunaan SSL atau autentikasi multifaktor (MFA). Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Syarat](#) dalam Panduan Pengguna IAM.

Menggunakan AWS IoT Analytics konsol

Untuk mengakses AWS IoT Analytics konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang AWS IoT Analytics sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksud untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan AWS IoT Analytics konsol, lampirkan juga kebijakan AWS terkelola berikut ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:BatchPutMessage",
        "iotanalytics:CancelPipelineReprocessing",
        "iotanalytics:CreateChannel",
```

```

        "iotanalytics:CreateDataset",
        "iotanalytics:CreateDatasetContent",
        "iotanalytics:CreateDatastore",
        "iotanalytics:CreatePipeline",
        "iotanalytics>DeleteChannel",
        "iotanalytics>DeleteDataset",
        "iotanalytics>DeleteDatasetContent",
        "iotanalytics>DeleteDatastore",
        "iotanalytics>DeletePipeline",
        "iotanalytics:DescribeChannel",
        "iotanalytics:DescribeDataset",
        "iotanalytics:DescribeDatastore",
        "iotanalytics:DescribeLoggingOptions",
        "iotanalytics:DescribePipeline",
        "iotanalytics:GetDatasetContent",
        "iotanalytics:ListChannels",
        "iotanalytics:ListDatasetContents",
        "iotanalytics:ListDatasets",
        "iotanalytics:ListDatastores",
        "iotanalytics:ListPipelines",
        "iotanalytics:ListTagsForResource",
        "iotanalytics:PutLoggingOptions",
        "iotanalytics:RunPipelineActivity",
        "iotanalytics:SampleChannelData",
        "iotanalytics:StartPipelineReprocessing",
        "iotanalytics:TagResource",
        "iotanalytics:UntagResource",
        "iotanalytics:UpdateChannel",
        "iotanalytics:UpdateDataset",
        "iotanalytics:UpdateDatastore",
        "iotanalytics:UpdatePipeline"
    ],
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:channel/
${channelName}",
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/
${datasetName}",
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:datastore/
${datastoreName}",
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:pipeline/
${pipelineName}"
    }
]
}

```

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Izinkan para pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada 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:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": [
        "arn:aws:iam::*:user/${aws:username}"
      ]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

Mengakses satu masukan AWS IoT Analytics

Dalam contoh ini, Anda ingin memberi pengguna Akun AWS akses ke salah satu AWS IoT Analytics saluran Anda `exampleChannel`. Anda juga ingin mengizinkan penggunaan untuk menambah, memperbarui, dan menghapus saluran.

Kebijakan memberikan `iotanalytics:ListChannels`, `iotanalytics:DescribeChannel`, `iotanalytics:CreateChannel`, `iotanalytics>DeleteChannel`, and `iotanalytics:UpdateChannel` izin kepada pengguna. Untuk contoh panduan untuk layanan Amazon S3 yang memberikan izin kepada pengguna dan mengujinya menggunakan konsol, [lihat Contoh panduan: Menggunakan kebijakan pengguna](#) untuk mengontrol akses ke bucket Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListChannelsInConsole",
      "Effect": "Allow",
      "Action": [
        "iotanalytics:ListChannels"
      ],
      "Resource": "arn:aws:iotanalytics::*"
    },
    {
      "Sid": "ViewSpecificChannelInfo",
      "Effect": "Allow",
      "Action": [
        "iotanalytics:DescribeChannel"
      ],
      "Resource": "arn:aws:iotanalytics:::exampleChannel"
    },
    {
      "Sid": "ManageChannels",
      "Effect": "Allow",
      "Action": [
        "iotanalytics:CreateChannel",
        "iotanalytics>DeleteChannel",
        "iotanalytics:DescribeChannel",
        "iotanalytics:ListChannels",

```

```

        "iotanalytics:UpdateChannel"
    ],
    "Resource": "arn:aws:iotanalytics:::exampleChannel/*"
}
]
}

```

Melihat AWS IoT Analytics saluran berdasarkan tag

Anda dapat menggunakan kondisi dalam kebijakan berbasis identitas untuk mengontrol akses ke AWS IoT Analytics sumber daya berdasarkan tag. Contoh ini menunjukkan bagaimana Anda dapat membuat kebijakan yang memungkinkan melihat `channel`. Namun, izin diberikan hanya jika `channel` tag `Owner` memiliki nilai nama pengguna pengguna tersebut. Kebijakan ini juga memberikan izin yang diperlukan untuk menyelesaikan tindakan ini di konsol.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListChannelsInConsole",
      "Effect": "Allow",
      "Action": "iotanalytics:ListChannels",
      "Resource": "*"
    },
    {
      "Sid": "ViewChannelsIfOwner",
      "Effect": "Allow",
      "Action": "iotanalytics:ListChannels",
      "Resource": "arn:aws:iotanalytics:*:*:channel/*",
      "Condition": {
        "StringEquals": {"iotanalytics:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

Anda dapat melampirkan kebijakan ini ke pengguna di akun Anda. Jika pengguna bernama `richard-roe` mencoba untuk melihat AWS IoT Analytics `channel`, `channel` harus ditandai `Owner=richard-roe` or `owner=richard-roe`. Jika tidak, aksesnya akan ditolak. Kunci tag kondisi `Owner` cocok dengan keduanya `Owner` dan `owner` karena nama kunci kondisi tidak

peka huruf besar/kecil. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.

Memecahkan masalah AWS IoT Analytics identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengannya AWS IoT Analytics.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di AWS IoT Analytics](#)
- [Saya tidak berwenang untuk melakukan iam:PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS IoT Analytics sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di AWS IoT Analytics

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda nama pengguna dan kata sandi Anda.

Contoh kesalahan berikut terjadi ketika mateojackson pengguna mencoba menggunakan konsol untuk melihat detail tentang channel tetapi tidak memiliki `iotanalytics:ListChannels` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotanalytics:``ListChannels`` on resource: ``my-example-channel``
```

Dalam hal ini, Mateo meminta administratornya memperbarui kebijakannya untuk memungkinkannya mengakses `my-example-channel` sumber daya menggunakan `iotanalytics:ListChannel` 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 Analytics.

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 Analytics. 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 saya Akun AWS untuk mengakses AWS IoT Analytics sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna 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 (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah AWS IoT Analytics mendukung fitur ini, lihat [Cara AWS IoT Analytics kerja 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 [Memberikan akses kepada pengguna eksternal yang sah \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan antara peran IAM dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Pencatatan dan pemantauan di AWS IoT Analytics

AWS menyediakan berbagai alat yang dapat Anda gunakan untuk memantau AWS IoT Analytics. Anda dapat mengonfigurasi beberapa alat ini agar melakukan pemantauan untuk Anda. Beberapa alat memerlukan intervensi manual. Kami menyarankan agar Anda mengotomatiskan tugas pemantauan sebanyak mungkin.

Alat pemantauan otomatis

Anda dapat menggunakan alat pemantauan otomatis berikut untuk melihat AWS IoT dan melaporkan saat terjadi kesalahan:

- Amazon CloudWatch Logs - Memantau, menyimpan, dan mengakses berkas log dari AWS CloudTrail atau sumber lainnya. Untuk informasi selengkapnya, lihat [Apa itu AWS CloudTrail](#) Memonitor File Log di Panduan CloudWatch Pengguna Amazon.
- AWS CloudTrail Pemantauan log - Bagikan berkas log antar akun, pantau berkas CloudTrail log secara langsung dengan mengirimnya ke CloudWatch Log, tulis aplikasi pemrosesan log di Java, dan validasi bahwa berkas log tidak berubah setelah pengiriman oleh CloudTrail. Untuk informasi selengkapnya, lihat [Bekerja dengan file CloudTrail log](#) di Panduan AWS CloudTrail Pengguna.

Alat pemantauan manual

Bagian penting lainnya dari pemantauan AWS IoT melibatkan pemantauan secara manual item yang tidak dicakup oleh alarm CloudWatch. Dasbor konsol AWS layanan lainnya memberikan at-a-glance tampilan keadaan AWS lingkungan Anda. AWS IoT CloudWatch. Sebaiknya Anda juga memeriksa berkas log AWS IoT Analytics.

- Konsol AWS IoT Analytics menampilkan:
 - Saluran
 - Alur
 - Penyimpanan data
 - Set data
 - Notebook
 - Pengaturan
 - Pelajari
- Tampilan CloudWatch beranda menunjukkan:

- Alarm dan status saat ini
- Grafik alarm dan sumber daya
- Status kesehatan layanan

Selain itu, Anda dapat menggunakan CloudWatch untuk melakukan hal berikut:

- Membuat [dasbor yang disesuaikan](#) untuk memantau layanan yang ingin Anda ketahui
- Grafik data metrik untuk memecahkan masalah dan menemukan tren
- Cari dan telusuri semua metrik sumber daya AWS Anda
- Membuat dan mengedit alarm untuk menerima notifikasi tentang masalah

Pemantauan dengan Amazon CloudWatch Logs

AWS IoT Analytics mendukung logging dengan Amazon CloudWatch. Anda dapat mengaktifkan dan mengonfigurasi CloudWatch pencatatan Amazon AWS IoT Analytics dengan menggunakan [operasi PutLoggingOptions API](#). Bagian ini menjelaskan bagaimana Anda dapat menggunakan PutLoggingOptions dengan AWS Identity and Access Management (IAM) untuk mengonfigurasi dan mengaktifkan CloudWatch pencatatan Amazon AWS IoT Analytics.

Untuk informasi lebih lanjut tentang CloudWatch Log, lihat [Panduan Pengguna Amazon CloudWatch Logs](#). Untuk informasi lebih lanjut tentang AWS IAM, lihat [Panduan AWS Identity and Access Management Pengguna](#).

Note

Sebelum Anda mengaktifkan AWS IoT Analytics logging, pastikan Anda memahami izin akses CloudWatch Log. Pengguna dengan akses ke CloudWatch Log dapat melihat informasi debugging Anda. Untuk informasi lebih lanjut, lihat [Kontrol Autentikasi dan akses untuk Amazon CloudWatch Logs](#).

Buat IAM role untuk mengaktifkan pencatatan log

Untuk membuat IAM role agar pencatatan log untuk Amazon CloudWatch

1. Gunakan [konsol AWS IAM](#) atau perintah AWS IAM CLI berikut, [CreateRole](#), untuk membuat peran IAM baru dengan kebijakan hubungan kepercayaan (kebijakan kepercayaan). Kebijakan kepercayaan memberi entitas, seperti Amazon CloudWatch, izin untuk mengambil peran.

```
aws iam create-role --role-name exampleRoleName --assume-role-policy-document
exampleTrustPolicy.json
```

exampleTrustPolicy.jsonFile berisi konten berikut.

Note

Contoh ini mencakup kunci konteks kondisi global untuk melindungi terhadap masalah keamanan wakil yang bingung. Ganti **123456789012** dengan IDAWS akun dan **aws-region** Anda denganAWS wilayahAWS sumber daya Anda. Untuk informasi selengkapnya, lihat [the section called “Pencegahan wakil bingung lintas layanan”](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:*"
        }
      }
    }
  ]
}
```

Anda menggunakan ARN peran ini nanti ketika Anda memanggilAWS IoT AnalyticsPutLoggingOptions perintah.

- GunakanAWS IAM [PutRolePolicy](#)untuk melampirkan kebijakan izin (a role policy) ke peran yang Anda buat di Langkah 1.

```
aws iam put-role-policy --role-name exampleRoleName --policy-name
examplePolicyName --policy-document exampleRolePolicy.json
```

exampleRolePolicyFile.json berisi konten berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

- Untuk memberikan AWS IoT Analytics izin untuk menempatkan peristiwa logging ke Amazon CloudWatch, gunakan CloudWatch perintah Amazon [PutResourcePolicy](#).

Note

Untuk membantu mencegah masalah keamanan wakil yang bingung, kami sarankan Anda menentukan `aws:SourceArn` dalam kebijakan sumber daya Anda. Ini membatasi akses untuk mengizinkan hanya permintaan yang berasal dari akun tertentu. Untuk informasi lebih lanjut tentang masalah deputi yang membingungkan, lihat [the section called "Pencegahan wakil bingung lintas layanan"](#).

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

exampleResourcePolicy.jsonFile berisi kebijakan sumber daya berikut.

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "iotanalytics.amazonaws.com"
        },
        "Action": "logs:PutLogEvents",
        "Resource": "*",
        "Condition": {
          "ArnLike": {
            "aws:SourceArn": "arn:aws:iotanalytics:us-east-1:123456789012:*/
* "
          },
          "StringEquals": {
            "aws:SourceAccount": "123456789012"
          }
        }
      }
    ]
  }

```

Mengonfigurasi dan mengaktifkan pencatatan log

Gunakan `PutLoggingOptions` perintah untuk mengonfigurasi dan mengaktifkan CloudWatch pencatatan Amazon untuk AWS IoT Analytics. `roleArnDiLoggingOptions` bidang harus berupa ARN dari peran yang Anda buat di bagian sebelumnya. Anda juga dapat menggunakan `DescribeLoggingOptions` perintah untuk memeriksa pengaturan opsi pencatatan log.

PutLoggingOptions

Menetapkan atau memperbarui opsi AWS IoT Analytics logging. Jika Anda memperbarui nilai `loggingOptions` bidang apa pun, dibutuhkan waktu hingga satu menit agar perubahan diterapkan. Selain itu, jika Anda mengubah kebijakan yang dilampirkan pada peran yang Anda tentukan di `roleArn` bidang (misalnya, untuk memperbaiki kebijakan yang tidak valid), diperlukan waktu hingga lima menit agar perubahan tersebut diterapkan. Untuk informasi selengkapnya, lihat [PutLoggingOptions](#).

DescribeLoggingOptions

Mengambil pengaturan saat ini dari opsi AWS IoT Analytics logging. Untuk informasi selengkapnya, lihat [DescribeLoggingOptions](#)

Namespace, metrik, dan dimensi

AWS IoT Analytics menempatkan metrik berikut ke dalam CloudWatch repositori Amazon:

Namespace
AWS/lotAnalitik

Metrik	Deskripsi
ActionExecution	Jumlah tindakan yang dilaksanakan.
ActionExecutionThrottled	Jumlah tindakan yang dicekik.
ActivityExecutionError	Jumlah kesalahan yang dihasilkan saat menjalankan aktivitas pipeline.
IncomingMessages	Jumlah pesan yang masuk ke saluran.
PipelineConcurrentExecutionCount	Jumlah kegiatan pipa, yang telah dijalankan secara bersamaan.

Dimensi	Deskripsi
ActionType	Jenis tindakan yang sedang dipantau.
ChannelName	Nama saluran yang sedang dipantau.
DatasetName	Nama set data yang sedang dipantau.
DatastoreName	Nama penyimpanan data yang sedang dipantau.
PipelineActivityName	Nama aktivitas alur yang sedang dipantau.
PipelineActivityType	Jenis aktivitas pipa yang sedang dipantau.
PipelineName	Nama dari alur yang sedang dipantau.

Pantau dengan CloudWatch Acara Amazon

AWS IoT Analytics secara otomatis menerbitkan peristiwa ke Amazon CloudWatch Events saat terjadi kesalahan waktu proses selama AWS Lambda aktivitas. Peristiwa ini berisi pesan kesalahan terperinci dan kunci objek Amazon Simple Storage Service (Amazon S3) yang menyimpan pesan saluran yang belum diproses. Anda dapat menggunakan kunci Amazon S3 untuk memproses ulang pesan saluran yang belum diproses. Untuk informasi selengkapnya [Pesan saluran](#), lihat, [StartPipelineReprocessing](#) API dalam Referensi AWS IoT Analytics API, dan [Apa Itu CloudWatch Acara Amazon](#) di Panduan Pengguna CloudWatch Acara Amazon.

Anda juga dapat mengonfigurasi target yang memungkinkan Amazon CloudWatch Events mengirim notifikasi atau mengambil tindakan lebih lanjut. Misalnya, Anda dapat mengirim notifikasi ke antrean Amazon Simple Queue Service (Amazon SQS), lalu memanggil `StartReprocessingMessage` API untuk memproses pesan saluran yang disimpan di objek Amazon S3. Amazon CloudWatch Events mendukung banyak jenis target, seperti berikut ini:

- Amazon Kinesis Streams
- AWS Lambda fungsi
- Topik Amazon Simple Notification Service (Amazon SNS)
- Antrean Amazon Simple Queue Service (Amazon SQS)

Untuk daftar target yang didukung, lihat [EventBridge Target Amazon](#) di Panduan EventBridge Pengguna Amazon.

Sumber daya CloudWatch Peristiwa Anda dan target terkait harus berada di AWS Wilayah di mana Anda membuat AWS IoT Analytics sumber daya. Untuk informasi selengkapnya, lihat [Endpoint dan kuota layanan](#) di bagian Referensi Umum AWS.

Notifikasi yang dikirim ke Amazon CloudWatch Events untuk kesalahan waktu proses dalam AWS Lambda aktivitas menggunakan format berikut.

```
{
  "version": "version-id",
  "id": "event-id",
  "detail-type": "IoT Analytics Pipeline Failure Notification",
  "source": "aws.iotanalytics",
  "account": "aws-account",
  "time": "timestamp",
```

```

"region": "aws-region",
"resources": [
  "pipeline-arn"
],
"detail": {
  "event-detail-version": "1.0",
  "pipeline-name": "pipeline-name",
  "error-code": "LAMBDA_FAILURE",
  "message": "error-message",
  "channel-messages": {
    "s3paths": [
      "s3-keys"
    ]
  },
  "activity-name": "lambda-activity-name",
  "lambda-function-arn": "lambda-function-arn"
}
}

```

Contoh pemberitahuan:

```

{
  "version": "0",
  "id": "204e672e-ef12-09af-4cfd-de3b53673ec6",
  "detail-type": "IoT Analytics Pipeline Failure Notification",
  "source": "aws.iotanalytics",
  "account": "123456789012",
  "time": "2020-10-15T23:47:02Z",
  "region": "ap-southeast-2",
  "resources": [
    "arn:aws:iotanalytics:ap-southeast-2:123456789012:pipeline/
test_pipeline_failure"
  ],
  "detail": {
    "event-detail-version": "1.0",
    "pipeline-name": "test_pipeline_failure",
    "error-code": "LAMBDA_FAILURE",
    "message": "Temp unavaliabile",
    "channel-messages": {
      "s3paths": [
        "test_pipeline_failure/channel/cmr_channel/__dt=2020-10-15
00:00:00/1602805530000_1602805560000_123456789012_cmr_channel_0_257.0.json.gz"
      ]
    }
  }
}

```

```
  },
  "activity-name": "LambdaActivity_33",
  "lambda-function-arn": "arn:aws:lambda:ap-
southeast-2:123456789012:function:lambda_activity"
}
}
```

Mendapatkan pemberitahuan data terlambat melalui Amazon CloudWatch Events

Saat Anda membuat konten set data menggunakan data dari kerangka waktu tertentu, beberapa data mungkin tidak tiba tepat waktu untuk diproses. Untuk memungkinkan penundaan, Anda dapat menentukan `deltaTimeOffsetQueryFilter` saat Anda [membuat kumpulan data](#) dengan menerapkan `queryAction` (kueri SQL). AWS IoT Analytics masih memproses data yang tiba dalam waktu delta, dan konten kumpulan data Anda memiliki jeda waktu. Fitur notifikasi data yang terlambat memungkinkan AWS IoT Analytics untuk mengirim notifikasi melalui [Amazon CloudWatch Events](#) saat data tiba setelah waktu delta.

Anda dapat menggunakan AWS IoT Analytics konsol, [API](#), [AWS Command Line Interface \(AWS CLI\)](#), atau [AWS SDK](#) untuk menentukan aturan data terlambat untuk kumpulan data.

Dalam AWS IoT Analytics API, `LateDataRuleConfiguration` objek mewakili pengaturan aturan data akhir dari dataset. Objek ini merupakan bagian dari `Dataset` objek yang terkait dengan operasi `CreateDataset` dan `UpdateDataset` API.

Parameter

Saat Anda membuat aturan data terlambat untuk kumpulan data dengan AWS IoT Analytics, Anda harus menentukan informasi berikut:

ruleConfiguration (LateDataRuleConfiguration)

Struktur yang berisi informasi konfigurasi aturan data akhir.

deltaTimeSessionWindowConfiguration

Struktur yang berisi informasi konfigurasi jendela sesi waktu delta.

[DeltaTime](#) menentukan interval waktu. Anda dapat menggunakan `DeltaTime` untuk membuat isi set data dengan data yang telah tiba di penyimpanan data sejak eksekusi terakhir. Sebagai contoh `DeltaTime`, lihat [Membuat set data SQL dengan jendela delta \(CLI\)](#).

timeoutInMinutes

Interval waktu. Anda dapat menggunakan `timeoutInMinutes` agar AWS IoT Analytics dapat mengumpulkan notifikasi data terlambat yang telah dihasilkan sejak eksekusi terakhir. AWS IoT Analytics mengirimkan satu batch pemberitahuan ke CloudWatch Acara sekaligus.

Tipe: Bilangan Bulat

Rentang yang valid: 1—60

ruleName

Nama aturan data akhir.

Jenis: String

Important

Untuk menentukan `LateDataRules`, kumpulan data harus menggunakan `DeltaTime` filter.

Konfigurasi aturan data terlambat (konsol)

Prosedur berikut ini menunjukkan cara mengonfigurasi aturan data terlambat dari kumpulan data di AWS IoT Analytics konsol.

Untuk mengonfigurasi aturan data terlambat

1. Masuk ke [konsol AWS IoT Analytics](#) tersebut.
2. Di panel navigasi, pilih Kumpulan data.
3. Di bawah Kumpulan data, pilih kumpulan data target.
4. Di panel navigasi, pilih Detail.
5. Di bagian jendela Delta, pilih Edit.
6. Pada Konfigurasi filter pemilihan data, lakukan hal berikut:
 - a. Untuk jendela Pemilihan data, pilih Waktu Delta.
 - b. Untuk Offset, masukkan periode waktu, lalu pilih unit.

- c. Untuk ekspresi Timestamp, masukkan ekspresi. Ini bisa berupa nama bidang timestamp atau ekspresi SQL yang dapat mengambil waktu, seperti `from_unixtime (time)`.

Untuk informasi selengkapnya tentang cara menulis ekspresi timestamp, lihat [Tanggal dan Waktu Fungsi dan Operator](#) di Dokumentasi Presto 0.172.

- d. Untuk Pemberitahuan data terlambat, pilih Aktif.
- e. Untuk waktu Delta, masukkan bilangan bulat. Rentang validnya adalah 1—60.
- f. Pilih Save (Simpan).

UPDATE DATA SET

Configure data selection filter

When creating a SQL data set, you can specify a deltaTime pre-filter to be applied to the message data to help limit the messages to those which have arrived since the last time the SQL data set content was created. [Learn more](#)

Data selection window

Delta time

Offset
Specifies possible latency in the arrival of a message

-3 Minutes

Timestamp expression

from_unixtime(time)

Late data notification
Enable late data notification to receive CloudWatch events if late data is detected.

Active

Delta time
IoT Analytics will emit a notification if late data is received within the value below

2 Minutes

[Back](#) [Save](#)

Konfigurasi aturan data terlambat (CLI)

Dalam AWS IoT Analytics API, `LateDataRuleConfiguration` objek mewakili pengaturan aturan data akhir dari dataset. Objek ini adalah bagian dari `Dataset` objek yang terkait dengan `CreateDataset` dan `UpdateDataset`. Anda dapat menggunakan [API](#), [AWS CLI](#), atau [AWSSDK](#) untuk menentukan aturan data terlambat untuk kumpulan data. Contoh berikut menggunakan AWS CLI.

Untuk membuat kumpulan data Anda dengan aturan data terlambat, jalankan perintah berikut. Perintah mengasumsikan bahwa `dataset.json` file tersebut dalam direktori saat ini.

Note

Anda dapat menggunakan [UpdateDataset](#) API untuk memperbarui kumpulan data yang ada.

```
aws iotanalytics create-dataset --cli-input-json file://dataset.json
```

`dataset.json` file harus berisi berikut:

- Ganti `demo_dataset` dengan nama set data target.
- Ganti `demo_datastore` dengan nama penyimpanan data target.
- Ganti `from_unixtime (waktu)` dengan nama bidang timestamp atau ekspresi SQL yang dapat mengambil waktu.

Untuk informasi selengkapnya tentang cara menulis ekspresi timestamp, lihat [Tanggal dan Waktu Fungsi dan Operator](#) di Dokumentasi Presto 0.172.

- Ganti `batas waktu` dengan integer antara 1-60.
- Ganti `demo_rule` dengan nama apapun.

```
{
  "datasetName": "demo_dataset",
  "actions": [
    {
      "actionName": "myDatasetAction",
      "queryAction": {
        "filters": [
```

```

        {
            "deltaTime": {
                "offsetSeconds": -180,
                "timeExpression": "from_unixtime(time)"
            }
        },
        "sqlQuery": "SELECT * FROM demo_datastore"
    }
},
"retentionPeriod": {
    "unlimited": false,
    "numberOfDays": 90
},
"lateDataRules": [
    {
        "ruleConfiguration": {
            "deltaTimeSessionWindowConfiguration": {
                "timeoutInMinutes": timeout
            }
        },
        "ruleName": "demo_rule"
    }
]
}

```

Berlangganan untuk menerima data notifikasi terlambat

Anda dapat membuat aturan di CloudWatch Acara yang menentukan cara memproses pemberitahuan data terlambat yang dikirim AWS IoT Analytics. Ketika CloudWatch Peristiwa menerima notifikasi, ia memanggil tindakan target yang ditentukan dalam aturan Anda.

Prasyarat untuk membuat aturan CloudWatch Events

Sebelum Anda membuat aturan CloudWatch Events untuk AWS IoT Analytics, Anda harus melakukan hal berikut:

- Biasakan diri Anda dengan peristiwa, dan target di CloudWatch Events.
- Buat dan konfigurasi [target](#) yang dipanggil oleh aturan CloudWatch Acara Anda. Aturan dapat memanggil berbagai jenis target, seperti berikut:
 - Amazon Kinesis Streams

- AWS Lambda fungsi
- Topik Amazon Simple Notification Service (Amazon SNS)
- Antrean Amazon Simple Queue Service (Amazon SQS)

Aturan CloudWatch Peristiwa Anda, dan target terkait harus berada diAWS Wilayah tempat Anda membuatAWS IoT Analytics sumber daya Anda. Untuk informasi selengkapnya, lihat [Endpoint dan kuota layanan](#) di bagian Referensi Umum AWS.

Untuk informasi lebih lanjut, lihat [Apa yang dimaksud dengan CloudWatch Events?](#) dan [Memulai CloudWatch Acara Amazon](#) di Panduan Pengguna CloudWatch Acara Amazon.

Kejadian pemberitahuan data terlambat

Acara untuk pemberitahuan data terlambat menggunakan format berikut.

```
{
  "version": "0",
  "id": "7f51dfa7-ffef-97a5-c625-abddbac5eadd",
  "detail-type": "IoT Analytics Dataset Lifecycle Notification",
  "source": "aws.iotanalytics",
  "account": "123456789012",
  "time": "2020-05-14T02:38:46Z",
  "region": "us-east-2",
  "resources": ["arn:aws:iotanalytics:us-east-2:123456789012:dataset/demo_dataset"],
  "detail": {
    "event-detail-version": "1.0",
    "dataset-name": "demo_dataset",
    "late-data-rule-name": "demo_rule",
    "version-ids": ["78244852-8737-4650-aa4d-3071a01338fa"],
    "message": null
  }
}
```

Membuat aturan CloudWatch Acara untuk menerima pemberitahuan data terlambat

Prosedur berikut ini menunjukkan cara membuat aturan yang mengirimkan notifikasi dataAWS IoT Analytics terlambat ke antrean Amazon SQS.

Untuk membuat aturan CloudWatch Peristiwa

1. Masuk ke [CloudWatchkonsol Amazon](#).

2. Di panel navigasi, di dalam Peristiwa, pilih Aturan.
3. Pada halaman Aturan, pilih Buat aturan.
4. Di bawah Sumber Peristiwa, pilih Pola Peristiwa.
5. Di Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan bagian, lakukan hal berikut:
 - a. Untuk Nama Layanan, pilih IoT Analytics
 - b. Untuk Jenis Acara, pilih Pemberitahuan Siklus Hidup Set Data IoT Analytics.
 - c. Pilih Nama kumpulan data spesifik, lalu masukkan nama kumpulan data target.
6. Di bawah Target, pilih Tambahkan target*.
7. Pilih antrian SQS, dan kemudian lakukan hal berikut:
 - Untuk Antrian*, pilih antrian target.
8. Pilih Konfigurasi detail.
9. Pada Langkah 2: Konfigurasi halaman detail aturan, masukkan nama dan deskripsi.
10. Pilih Buat aturan.

Mencatat panggilan API AWS IoT Analytics dengan AWS CloudTrail

AWS IoT Analytics terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS IoT Analytics. CloudTrail menangkap subset panggilan API untuk AWS IoT Analytics sebagai peristiwa, termasuk panggilan dari AWS IoT Analytics konsol tersebut dan dari panggilan kode ke AWS IoT Analytics API. Jika membuat jejak, Anda dapat mengaktifkan pengiriman tindakan CloudTrail berkelanjutan ke bucket Amazon S3, termasuk peristiwa untuk AWS IoT Analytics. Jika Anda tidak dapat mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol di Riwayat peristiwa. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat AWS IoT Analytics, alamat IP asal permintaan dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjut CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Informasi AWS IoT Analytics di AWS CloudTrail

CloudTrail diaktifkan pada AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di AWS IoT Analytics, aktivitas tersebut dicatat dalam CloudTrail peristiwa bersama peristiwa AWS layanan

lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi lebih lanjut, lihat [Melihat peristiwa dengan riwayat CloudTrail peristiwa](#).

Untuk catatan berkelanjutan tentang peristiwa di akun AWS Anda, termasuk peristiwa untuk AWS IoT Analytics, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah . Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lainnya untuk dianalisis lebih lanjut dan bertindak berdasarkan data kejadian yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, lihat :

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail Layanan yang didukung dan integrasi](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima berkas CloudTrail log dari beberapa wilayah](#) dan [Menerima berkas CloudTrail log dari beberapa akun](#)

AWS IoT Analytics mendukung pencatatan tindakan berikut sebagai peristiwa dalam berkas CloudTrail log:

- [CancelPipelineReprocessing](#)
- [CreateChannel](#)
- [CreateDataset](#)
- [CreateDatasetContent](#)
- [CreateDatastore](#)
- [CreatePipeline](#)
- [DeleteChannel](#)
- [DeleteDataset](#)
- [DeleteDatasetContent](#)
- [DeleteDatastore](#)
- [DeletePipeline](#)
- [DescribeChannel](#)
- [DescribeDataset](#)
- [DescribeDatastore](#)

- [DescribeLoggingOptions](#)
- [DescribePipeline](#)
- [GetDatasetContent](#)
- [ListChannels](#)
- [ListDatasets](#)
- [ListDatastores](#)
- [ListPipelines](#)
- [PutLoggingOptions](#)
- [RunPipelineActivity](#)
- [SampleChannelData](#)
- [StartPipelineReprocessing](#)
- [UpdateChannel](#)
- [UpdateDataset](#)
- [UpdateDatastore](#)
- [UpdatePipeline](#)

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau AWS Identity and Access Management.
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#) .

Memahami entri file log AWS IoT Analytics

Penjejak adalah konfigurasi yang dapat membuat pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang telah Anda tentukan. CloudTrail berkas log berisi satu atau beberapa entri log. Sebuah peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya.

CloudTrail berkas log bukan jejak tumpukan panggilan API publik yang berurutan, sehingga berkas log tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateChannel` tindakan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDE12345FGHIJ67890B:AnalyticsChannelTestFunction",
    "arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/AnalyticsChannelTestFunction",
    "accountId": "123456789012",
    "accessKeyId": "ABCDE12345FGHIJ67890B",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-02-14T23:43:12Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ABCDE12345FGHIJ67890B",
        "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
        "accountId": "123456789012",
        "userName": "AnalyticsRole"
      }
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "ABCDE12345FGHIJ67890B",
      "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
      "accountId": "123456789012",
      "userName": "AnalyticsRole"
    }
  },
  "eventTime": "2018-02-14T23:55:14Z",
  "eventSource": "iotanalytics.amazonaws.com",
  "eventName": "CreateChannel",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "198.162.1.0",
  "userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
  "requestParameters": {
    "channelName": "channel_channeltest"
  },
  "responseElements": {
    "retentionPeriod": {
      "unlimited": true
    }
  },
  "channelName": "channel_channeltest",
  "channelArn": "arn:aws:iotanalytics:us-east-1:123456789012:channel/channel_channeltest"
},
```

```
"requestID": "7f871429-11e2-11e8-9eee-0781b5c0ac59",
"eventID": "17885899-6977-41be-a6a0-74bb95a78294",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateDataset` tindakan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDE12345FGHIJ67890B:AnalyticsDatasetTestFunction",
    "arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/AnalyticsDatasetTestFunction",
    "accountId": "123456789012",
    "accessKeyId": "ABCDE12345FGHIJ67890B",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-02-14T23:41:36Z"
      }
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "ABCDE12345FGHIJ67890B",
      "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
      "accountId": "123456789012",
      "userName": "AnalyticsRole"
    }
  },
  "eventTime": "2018-02-14T23:53:39Z",
  "eventSource": "iotanalytics.amazonaws.com",
  "eventName": "CreateDataset",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "198.162.1.0",
  "userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
  "requestParameters": {
    "datasetName": "dataset_datasettest"
  },
  "responseElements": {
    "datasetArn": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/dataset_datasettest",
  }
}
```

```
"datasetName": "dataset_datasettest"
},
"requestID": "46ee8dd9-11e2-11e8-979a-6198b668c3f0",
"eventID": "5abe21f6-ee1a-48ef-afc5-c77211235303",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

Validasi kepatuhan untuk AWS IoT Analytics

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. 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 Layanan AWS 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](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut

Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).

- [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](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [AWS Audit Manager](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di AWS IoT Analytics

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Kawasan menyediakan beberapa Zona Ketersediaan yang terpisah dan terisolasi secara fisik, yang tersambung dengan jejaring jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Availability Zones, Anda dapat merancang dan mengoperasikan aplikasi dan database yang secara otomatis gagal di antara Availability Zones tanpa gangguan. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripada infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [infrastruktur AWS global](#).

Keamanan infrastruktur di AWS IoT Analytics

Sebagai layanan terkelola, AWS IoT Analytics 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 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 dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Kuota AWS IoT Analytics

Referensi Umum AWS Panduan ini menyediakan kuota default AWS IoT Analytics untuk AWS akun. Kecuali ditentukan, masing-masing kuota adalah per AWS Wilayah. Untuk informasi selengkapnya, lihat [AWS IoT Analytics titik akhir dan kuota](#) serta [kuota AWS layanan](#) di Referensi Umum AWS Panduan.

Untuk meminta peningkatan kuota layanan, kirim kasus Support di konsol [Pusat Dukungan](#). Untuk informasi lebih lanjut, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

AWS IoT Analytics perintah

Baca topik ini untuk mempelajari tentang operasi API AWS IoT Analytics, termasuk permintaan sampel, tanggapan, dan kesalahan untuk protokol layanan web yang didukung.

Tindakan AWS IoT Analytics

Anda dapat menggunakan AWS IoT Analytics Perintah API untuk mengumpulkan, memproses, menyimpan, dan menganalisis data IoT Anda. Untuk informasi selengkapnya, lihat [tindakan](#) yang didukung oleh AWS IoT Analytics di [AWS IoT Analytics Referensi API](#).

Parameter [AWS IoT Analytics](#) [seksi](#) di [AWS CLI Referensi Perintah](#) termasuk AWS CLI perintah yang dapat Anda gunakan untuk mengelola dan memanipulasi AWS IoT Analytics.

AWS IoT Analytics data

Anda dapat menggunakan AWS IoT Analytics Perintah API Data untuk melakukan aktivitas lanjutan dengan AWS IoT Analytics `channel`, `pipeline`, `datastore`, dan `dataset`. Untuk informasi selengkapnya, lihat [tipe data](#) yang didukung oleh AWS IoT Analytics Data di [AWS IoT Analytics Referensi API](#).

Pemecahan Masalah AWS IoT Analytics

Lihat bagian berikut untuk memecahkan masalah kesalahan dan menemukan dan solusi yang mungkin untuk menyelesaikan masalah dengan AWS IoT Analytics.

Topik

- [Bagaimana cara saya tahu jika pesan saya masuk AWS IoT Analytics?](#)
- [Mengapa pipeline saya kehilangan pesan? Bagaimana cara saya memperbaikinya?](#)
- [Mengapa tidak ada data di penyimpanan data saya?](#)
- [Mengapa dataset saya hanya ditampilkan __dt?](#)
- [Bagaimana cara membuat kode peristiwa yang didorong oleh penyelesaian kumpulan data?](#)
- [Bagaimana cara mengkonfigurasi instance notebook saya dengan benar untuk digunakan AWS IoT Analytics?](#)
- [Mengapa saya tidak bisa membuat notebook dalam sebuah instance?](#)
- [Mengapa saya tidak melihat kumpulan data saya di Amazon QuickSight?](#)
- [Mengapa saya tidak melihat tombol containerize pada Notebook Jupyter saya yang ada?](#)
- [Mengapa instalasi plugin containerization saya gagal?](#)
- [Mengapa plugin containerization saya melempar kesalahan?](#)
- [Mengapa saya tidak melihat variabel saya selama penampung?](#)
- [Variabel apa yang dapat saya tambahkan ke wadah saya sebagai input?](#)
- [Bagaimana cara mengatur output kontainer saya sebagai masukan untuk analisis selanjutnya?](#)
- [Mengapa dataset kontainer saya gagal?](#)

Bagaimana cara saya tahu jika pesan saya masuk AWS IoT Analytics?

Periksa apakah aturan untuk menyuntikkan data ke saluran melalui aturan-mesin dikonfigurasi dengan benar.

```
aws iot get-topic-rule --rule-name your-rule-name
```

Responsnya akan terlihat seperti berikut.


```
{
  "ruleArn": "arn:aws:iot:us-west-2:your-account-id:rule/your-rule-name",
  "rule": {
    "awsIotSqlVersion": "2016-03-23",
    "sql": "SELECT * FROM 'iot/your-rule-name'",
    "ruleDisabled": false,
    "actions": [
      {
        "iotAnalytics": {
          "channelArn":
            "arn:aws:iotanalytics:region:your_account_id:channel/your-channel-name"
        }
      }
    ],
    "ruleName": "your-rule-name"
  }
}
```

Pastikan wilayah dan nama saluran yang digunakan dalam aturan sudah benar. Untuk memastikan data Anda mencapai mesin aturan dan aturan dijalankan dengan benar, Anda mungkin ingin menambahkan target baru untuk menyimpan pesan masuk di bucket Amazon S3 sementara.

Mengapa pipeline saya kehilangan pesan? Bagaimana cara cara saya memperbaikinya?

- Aktivitas telah menerima masukan JSON yang tidak valid:

Semua aktivitas, kecuali aktivitas Lambda, secara khusus memerlukan string JSON yang valid sebagai masukan. Jika JSON yang diterima oleh suatu aktivitas tidak valid, maka pesan akan dijatuhkan dan tidak masuk ke penyimpanan data. Pastikan Anda memasukkan pesan JSON yang valid ke dalam layanan. Dalam kasus input biner, pastikan aktivitas pertama dalam pipeline Anda adalah aktivitas Lambda yang mengubah data biner menjadi JSON yang valid sebelum meneruskannya ke aktivitas berikutnya atau menyimpannya di penyimpanan data. Untuk informasi selengkapnya, lihat [Contoh fungsi Lambda 2](#).

- Fungsi Lambda yang dipanggil oleh aktivitas Lambda memiliki izin yang tidak mencukupi:

Pastikan bahwa setiap fungsi Lambda dalam aktivitas Lambda memiliki izin untuk dipanggil dari AWS IoT Analytics layanan. Anda dapat menggunakan AWS CLI perintah berikut untuk memberikan izin.

```
aws lambda add-permission --function-name <name> --region <region> --statement-id <id> --principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
```

- Aktivitas filter atau `removeAttribute` tidak didefinisikan dengan benar:

Pastikan definisi jika ada `filter` atau `removeAttribute` aktivitas yang benar. Jika Anda menyaring pesan atau menghapus semua atribut dari pesan, pesan tersebut tidak ditambahkan ke penyimpanan data.

Mengapa tidak ada data di penyimpanan data saya?

- Ada penundaan antara konsumsi data dan ketersediaan data:

Mungkin diperlukan beberapa menit setelah data dicerna ke saluran sebelum data tersebut tersedia di penyimpanan data. Waktunya bervariasi berdasarkan jumlah aktivitas pipeline dan definisi aktivitas Lambda khusus apa pun di pipeline Anda.

- Pesan sedang disaring di pipeline Anda:

Pastikan Anda tidak menjatuhkan pesan di pipeline. (Lihat pertanyaan dan tanggapan sebelumnya.)

- Kueri kumpulan data Anda salah:

Pastikan kueri yang menghasilkan dataset dari penyimpanan data sudah benar. Hapus filter yang tidak perlu dari kueri untuk memastikan data Anda mencapai penyimpanan data Anda.

Mengapa dataset saya hanya ditampilkan `__dt`?

- Kolom ini ditambahkan oleh layanan secara otomatis dan berisi perkiraan waktu konsumsi data. Ini dapat digunakan untuk mengoptimalkan pertanyaan Anda. Jika kumpulan data Anda tidak berisi apa pun selain ini, lihat pertanyaan dan tanggapan sebelumnya.

Bagaimana cara membuat kode peristiwa yang didorong oleh penyelesaian kumpulan data?

- Anda harus mengatur polling berdasarkan `describe-dataset` perintah untuk memeriksa apakah status dataset dengan stempel waktu tertentu SUCCEEDED.

Bagaimana cara mengkonfigurasi instance notebook saya dengan benar untuk digunakan AWS IoT Analytics?

Ikuti langkah-langkah berikut untuk memastikan peran IAM yang Anda gunakan untuk membuat instance notebook memiliki izin yang diperlukan:

1. Pergi ke SageMaker konsol dan buat instance notebook.
2. Isi detailnya dan pilih buat peran baru. Perhatikan ARN peran.
3. Buat instans notebook. Ini juga menciptakan peran yang SageMaker dapat digunakan.
4. Buka konsol IAM dan ubah SageMaker peran yang baru dibuat. Ketika Anda membuka peran itu, itu harus memiliki kebijakan yang dikelola.
5. Klik tambahkan kebijakan sebaris, pilih IoTAnalytics sebagai layanan, dan di bawah izin baca, pilih `GetDatasetContent`.
6. Tinjau kebijakan, tambahkan nama kebijakan, lalu buat kebijakan. Peran yang baru dibuat sekarang memiliki izin kebijakan untuk membaca dataset dari AWS IoT Analytics.
7. Pergi ke AWS IoT Analytics konsol dan buat notebook di instance notebook.
8. Tunggu instance notebook berada dalam keadaan "In Service".
9. Pilih buat notebook, dan pilih instance notebook yang Anda buat. Ini menciptakan notebook Jupyter dengan template yang dipilih yang dapat mengakses dataset Anda.

Mengapa saya tidak bisa membuat notebook dalam sebuah instance?

- Pastikan Anda membuat instance notebook dengan kebijakan IAM yang benar. (Ikuti langkah-langkah dalam pertanyaan sebelumnya.)

- Pastikan instance notebook berada dalam keadaan “In Service”. Ketika Anda membuat sebuah instance, itu dimulai dalam keadaan “Pending”. Biasanya dibutuhkan sekitar lima menit untuk masuk ke keadaan “In Service”. Jika instance notebook masuk ke status “Gagal” setelah sekitar lima menit, periksa izin lagi.

Mengapa saya tidak melihat kumpulan data saya di Amazon QuickSight?

Amazon QuickSight mungkin memerlukan izin untuk membaca konten AWS IoT Analytics kumpulan data Anda. Untuk memberikan izin, ikuti langkah-langkah ini.

1. Pilih nama akun Anda di sudut kanan atas Amazon QuickSight dan pilih Kelola QuickSight.
2. Di panel navigasi sebelah kiri, pilih Keamanan & izin. Di bawah QuickSight akses ke AWS layanan, verifikasi bahwa akses diberikan kepada AWS IoT Analytics.
 - a. Jika AWS IoT Analytics tidak memiliki akses, pilih Tambah atau hapus.
 - b. Pilih kotak di samping AWS IoT Analytics dan kemudian pilih Perbarui. Ini memberi QuickSight izin Amazon untuk membaca konten kumpulan data Anda.
3. Coba lagi untuk memvisualisasikan data Anda.

Pastikan Anda memilih AWS Wilayah yang sama untuk keduanya AWS IoT Analytics dan Amazon QuickSight. Jika tidak, Anda mungkin mengalami masalah dalam mengakses AWS sumber daya. Untuk daftar Wilayah yang didukung, lihat [AWS IoT Analytics titik akhir dan kuota](#) serta [QuickSight titik akhir dan kuota Amazon](#) di bagian Referensi Umum Amazon Web Services.

Mengapa saya tidak melihat tombol containerize pada Notebook Jupyter saya yang ada?

- Hal ini disebabkan oleh Plugin AWS IoT Analytics Containerization yang hilang. Jika Anda membuat contoh SageMaker notebook Anda sebelum 23 Agustus 2018, Anda perlu menginstal plugin secara manual dengan mengikuti petunjuk di [Containerizing notebook](#).
- Jika Anda tidak melihat tombol containerize setelah membuat instance SageMaker notebook dari AWS IoT Analytics konsol atau menginstalnya secara manual, hubungi dukungan AWS IoT Analytics teknis.

Mengapa instalasi plugin containerization saya gagal?

- Biasanya, instalasi plugin gagal karena izin yang hilang dalam contoh SageMaker notebook. Untuk izin yang diperlukan untuk instance notebook, lihat [Izin](#) dan tambahkan izin yang diperlukan ke peran instance notebook. Jika masalah berlanjut, buat instance notebook baru dari AWS IoT Analytics konsol.
- Anda dapat dengan aman mengabaikan pesan berikut di log jika muncul selama instalasi plugin: “Untuk menginisialisasi ekstensi ini di browser setiap kali notebook (atau aplikasi lain) dimuat.”

Mengapa plugin containerization saya melempar kesalahan?

- Containerization dapat gagal dan menghasilkan kesalahan karena berbagai alasan. Pastikan Anda menggunakan kernel yang benar sebelum menyusun notebook Anda. Kernel dalam wadah dimulai dengan awalan “Kontainer”.
- Karena plugin membuat dan menyimpan gambar docker di repositori ECR, pastikan bahwa peran instance notebook Anda memiliki izin yang cukup untuk membaca, membuat daftar, dan membuat repositori ECR. Untuk izin yang diperlukan untuk instance notebook, lihat [Izin](#) dan tambahkan izin yang diperlukan ke peran instance notebook.
- Pastikan juga bahwa nama repositori sesuai dengan persyaratan ECR. Nama repositori ECR harus dimulai dengan huruf dan hanya dapat berisi huruf kecil, angka, tanda hubung, garis bawah, dan garis miring.
- Jika proses containerization gagal dengan kesalahan: "Instance ini memiliki ruang kosong yang tidak mencukupi untuk menjalankan containerization" coba gunakan instance yang lebih besar untuk menyelesaikan masalah.
- Jika Anda melihat kesalahan koneksi atau kesalahan pembuatan gambar, coba lagi. Jika masalah berlanjut, restart instance dan instal versi plugin terbaru.

Mengapa saya tidak melihat variabel saya selama penampung?

- Plugin AWS IoT Analytics containerization secara otomatis mengenali semua variabel di notebook Anda setelah menjalankan notebook dengan kernel “Containerized”. Gunakan salah satu kernel dalam peti kemas untuk menjalankan notebook, lalu lakukan kontainerisasi.

Variabel apa yang dapat saya tambahkan ke wadah saya sebagai input?

- Anda dapat menambahkan variabel apa pun yang nilainya ingin Anda modifikasi selama runtime sebagai masukan ke wadah Anda. Hal ini memungkinkan Anda untuk menjalankan wadah yang sama dengan parameter yang berbeda yang perlu disediakan pada saat pembuatan dataset. Plugin AWS IoT Analytics containerization Jupyter menyederhanakan proses ini dengan secara otomatis mengenali variabel di notebook dan membuatnya tersedia sebagai bagian dari proses containerization.

Bagaimana cara mengatur output kontainer saya sebagai masukan untuk analisis selanjutnya?

- Lokasi S3 spesifik tempat artefak yang dieksekusi dapat disimpan dibuat untuk setiap rangkaian data kontainer Anda. Untuk mengakses lokasi keluaran ini, buat variabel dengan tipe `outputFileUriValue` dataset kontainer Anda. Nilai variabel ini harus berupa jalur S3 yang digunakan untuk menyimpan file keluaran tambahan Anda. Untuk mengakses artefak yang disimpan ini dalam proses selanjutnya, Anda dapat menggunakan `getDatasetContent` API dan memilih file keluaran yang sesuai yang diperlukan untuk proses selanjutnya.

Mengapa dataset kontainer saya gagal?

- Pastikan Anda meneruskan yang benar `executionRole` ke set data kontainer. Kebijakan kepercayaan dari `executionRole` harus mencakup keduanya `iotanalytics.amazonaws.com` dan `sagemaker.amazonaws.com`.
- Jika Anda melihat `AlgorithmError` sebagai alasan kegagalan, cobalah untuk men-debug kode kontainer Anda secara manual. Hal ini terjadi jika ada bug dalam kode kontainer atau peran eksekusi tidak memiliki izin untuk mengeksekusi kontainer. Jika Anda melakukan kontainer dengan menggunakan plugin AWS IoT Analytics Jupyter, buat instance SageMaker notebook baru dengan peran yang sama dengan `ExecutionRole` dari `ContainerDataSet` dan coba jalankan notebook secara manual. Jika wadah dibuat di luar plugin Jupyter, coba jalankan kode secara manual dan batasi izin ke `ExecutionRole`.

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada AWS IoT Analytics Panduan Pengguna setelah Selasa, 03 November 2020. Untuk informasi lebih lanjut tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke feed RSS.

Perubahan	Deskripsi	Tanggal
Peluncuran wilayah	AWS IoT Analytics kini tersedia di wilayah Asia Pacific (Mumbai)	18 Agustus, 2021
Kueri dengan JOIN	Pembaruan ini memungkinkan Anda untuk menggunakan JOIN kueri AWS IoT Analytics set data.	27 Juli, 2021
Integrasi dengan AWS IoT SiteWise	Sekarang Anda dapat menggunakan AWS IoT Analytics kueri AWS IoT SiteWise data.	27 Juli, 2021
Partisi khusus	AWS IoT Analytics sekarang umumnya mendukung partisi data Anda sesuai dengan atribut pesan atau atribut yang ditambahkan melalui kegiatan pipa.	14 Juni 2021
Memproses ulang pesan saluran	Pembaruan ini memungkinkan Anda memproses ulang data saluran di objek Amazon S3 yang ditentukan.	15 Desember 2020
Skema Parquet	AWS IoT Analytics menyimpan data sekarang mendukung format file Parquet.	15 Desember 2020

Pemantauan dengan CloudWatch Peristiwa	AWS IoT Analytics secara otomatis memublikasikan acara ke Amazon CloudWatch Peristiwa ketika kesalahan runtime terjadi selama AWS Lambda aktivitas.	15 Desember 2020
Pemberitahuan data terlambat	Anda dapat menggunakan fitur ini untuk menerima pemberitahuan melalui Amazon CloudWatch Peristiwa saat data terlambat tiba.	9 November 2020
Peluncuran wilayah	Luncurkan AWS IoT Analytics di China (Beijing).	4 November 2020

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting pada AWS IoT Analytics Panduan Pengguna sebelum 4 November 2020.

Perubahan	Deskripsi	Tanggal
Peluncuran wilayah	Luncurkan AWS IoT Analytics di Wilayah Asia Pacific (Sydney)	16 Juli 2020
Pembaruan	Reorganisasi dokumentasi.	07 Mei 2020

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.