



AWS Panduan Keputusan

# Memilih layanan integrasi AWS aplikasi



# Memilih layanan integrasi AWS aplikasi: AWS Panduan Keputusan

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak dapat digunakan sehubungan dengan produk atau layanan yang bukan milik Amazon, dalam bentuk apa pun yang mungkin menimbulkan kebingungan di kalangan pelanggan, atau dalam bentuk apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon adalah properti dari pemiliknya masing-masing, yang mungkin atau mungkin tidak berafiliasi dengan, berhubungan dengan, atau disponsori oleh Amazon.

---

# Table of Contents

Panduan keputusan .....	i
Pengantar .....	1
Memahami .....	2
Mendefinisikan .....	3
Pertimbangkan .....	5
Pilih .....	7
Gunakan .....	11
Jelajahi .....	19
Riwayat dokumen .....	21
.....	xxii

# Memilih layanan integrasi AWS aplikasi

## Mengambil langkah pertama

Tujuan	Bantu menentukan layanan integrasi AWS aplikasi mana yang paling cocok untuk beban kerja Anda.
Terakhir diperbarui	April 16, 2025
Layanan yang tercakup	<ul style="list-style-type: none"><li>• <a href="#">Amazon SNS</a></li><li>• <a href="#">Amazon SQS</a></li><li>• <a href="#">Amazon EventBridge</a></li><li>• <a href="#">Amazon MQ</a></li><li>• <a href="#">Amazon Kinesis Data Streams</a></li><li>• <a href="#">Amazon MSK</a></li><li>• <a href="#">Amazon MWAA</a></li><li>• <a href="#">AWS Step Functions</a></li></ul>

## Pengantar

Integrasi aplikasi adalah rangkaian layanan yang memungkinkan komunikasi antara komponen yang dipisahkan dalam layanan mikro, sistem terdistribusi, dan aplikasi tanpa server. Amazon Web Services (AWS) menawarkan lebih dari setengah lusin layanan integrasi aplikasi untuk mendukung beragam beban kerja yang berjalan di cloud.

Memilih layanan integrasi yang paling cocok untuk organisasi dan beban kerja Anda bisa menjadi sulit. Panduan keputusan ini akan membantu Anda mengajukan pertanyaan yang tepat untuk menemukan kebutuhan Anda dan memberikan panduan yang jelas tentang cara mengevaluasi dan memilih layanan integrasi yang tepat untuk beban kerja Anda.

[Klip 8½ menit ini berasal dari rekaman presentasi satu jam oleh AWS direktur strategi perusahaan Gregor Hohpe di re:Invent 2022. AWS Ini memberikan gambaran umum tentang layanan integrasi AWS aplikasi yang tersedia.](#)

# Memahami

Salah satu manfaat utama modernisasi adalah kemampuan untuk mengalihkan tanggung jawab operasional, memungkinkan Anda membebaskan sumber daya untuk melakukan lebih banyak aktivitas yang didorong oleh nilai tambah dan inovasi.

Ada spektrum opsi tanggung jawab bersama di berbagai tingkat modernisasi, mulai dari hosting broker pesan Anda di Amazon Elastic Compute Cloud (Amazon EC2) —tempat Anda mengelola penskalaan, konfigurasi keamanan, penyediaan, penambalan, dan lainnya—hingga penawaran tanpa server di mana semua infrastruktur dasar dikelola.



Saat Anda mulai menjelajahi dan memahami kriteria, lingkungan, dan rangkaian layanan integrasi yang AWS ditawarkan, kami sarankan Anda meninjau beberapa praktik terbaik. Praktik terbaik ini berlaku terlepas dari layanan mana (atau rangkaian layanan) yang Anda pilih.

## Memahami integrasi di lingkungan Anda

Sudah umum bahwa beberapa organisasi menghabiskan lebih banyak waktu daripada yang mereka inginkan untuk mempertahankan integrasi sumber terbuka. Kami menyarankan Anda

mempertimbangkan sumber komunitas, and/or dukungan dari perusahaan atau yayasan saat melakukan investasi ini. Investasi dalam proyek-proyek ini tidak hanya finansial, tetapi juga investasi dalam modal pengetahuan dan potensi utang teknis, karena komponen dan integrasi terkait ini biasanya perlu diperbarui. Untuk informasi selengkapnya, lihat [blog AWS Open Source](#).

## Pahami karakteristik arsitektur Anda

Kemampuan untuk mendukung berbagai arsitektur adalah penting. Kami menyarankan Anda menggunakan [AWS Well-Architected](#) Framework sebagai panduan untuk membantu Anda memahami keputusan yang Anda buat saat membangun arsitektur. AWS Selain itu, menggunakan Well-Architected Framework memungkinkan Anda mempelajari praktik terbaik arsitektur untuk merancang dan mengoperasikan sistem yang andal, terukur, aman, efisien, dan hemat biaya di Cloud.

## Gunakan kombinasi layanan integrasi

Jika Anda menggunakan layanan yang dibuat khusus, kombinasi layanan mungkin paling cocok untuk kasus penggunaan Anda. Berikut ini mencantumkan beberapa cara umum AWS pelanggan menggunakan kombinasi layanan.

- Merutekan peristiwa Amazon EventBridge atau Amazon Simple Notification Service (Amazon SNS) ke antrian Amazon Simple Queue Service (Amazon SQS) sebagai buffer untuk konsumen hilir.
- Menarik peristiwa langsung dari aliran (Amazon Kinesis Data Streams atau Amazon Managed Streaming untuk Apache Kafka (Amazon MSK)) atau antrian (Amazon SQS atau EventBridge Amazon MQ) dengan Pipes dan mengirim acara ke bus untuk mendorong konsumen. EventBridge
- Perutean EventBridge atau peristiwa Amazon SNS ke Kinesis Data Streams atau Amazon MSK untuk mengumpulkan dan melihat analitik.

## Mendefinisikan

Setelah Anda memiliki gambaran yang lebih jelas tentang kriteria, lingkungan, arah strategis, dan layanan yang tersedia (termasuk modalitas penyebaran yang dihosting dan dikelola), Anda perlu mengidentifikasi persyaratan integrasi Anda. Anda mungkin sudah mengetahui beberapa persyaratan Anda jika Anda bermigrasi ke platform integrasi atau broker pesan yang ada. Namun, Anda perlu menetapkan bagaimana persyaratan ini akan berubah jika Anda pindah ke lingkungan cloud, jika sama sekali.

## Platform perpesanan atau streaming

Platform ini diharapkan dapat memenuhi fungsionalitas bisnis tertentu. Gunakan contoh kasus penggunaan berikut saat mempertimbangkan fungsi mana yang Anda perlukan.

### Contoh 1:

Pertimbangkan perusahaan asuransi yang menerima klaim berbeda sebagai pesan untuk jenis klaim yang berbeda (auto, rumah, atau jiwa) dengan aturan bisnis yang berbeda. Ini mungkin berarti bahwa konsumen pesan harus memiliki fungsionalitas untuk merutekan klaim ke tujuan yang berbeda berdasarkan properti header dalam pesan.

### Contoh 2:

Pertimbangkan maskapai penerbangan di mana pembaruan status penerbangan perlu memberi tahu semua sistem yang terhubung seperti operasi bagasi atau gerbang menggunakan protokol seperti Advanced Messaging Queuing Protocol (AMQP). Pertanyaan besar dengan primitif kasus penggunaan fungsional dan bisnis adalah apa yang merupakan platform perpesanan yang paling sesuai. Kami memiliki beberapa pilihan yang dapat menentukan kesesuaian platform berdasarkan kasus penggunaan.

- **Adopsi pasar:** Platform ini diadopsi secara luas oleh komunitas pelanggan yang besar dan cukup cocok untuk sebagian besar kasus penggunaan. Ini dicoba dan diuji dengan komunitas dukungan yang dinamis untuk masalah apa pun yang mungkin dihadapi. Ini adalah keputusan berisiko rendah dengan pelatihan yang cukup tersedia untuk sumber daya pengembangan.
- **Paling cocok untuk kasus penggunaan:** Platform ini akan disesuaikan untuk kasus penggunaan industri tertentu seperti maskapai penerbangan, logistik, atau perawatan kesehatan. Mereka mungkin paling cocok untuk kasus penggunaan tersebut dengan templat siap pakai yang tersedia untuk diadopsi. Platform ini bisa mudah untuk memulai tetapi tidak memiliki tingkat adopsi di pasar serta fleksibilitas. Mengadopsi jenis platform ini mungkin memerlukan waktu dan sumber daya yang luas untuk validasi dan membangun keahlian di rumah.
- **Modern:** Platform ini dibangun dengan arsitektur generasi berikutnya untuk mengatasi penerapan skala cloud, multi-tenancy, pemulihan bencana, dan jenis harga tanpa server. Menggunakan jenis platform ini mungkin memerlukan beberapa refactoring beban kerja untuk kelangsungan hidup jangka panjang. Ini menggunakan platform cloud-native dan berfokus pada penggunaan prinsip-prinsip aplikasi modern yang dirancang dengan baik.

### Contoh 3:

Jika platform perpesanan adalah bagian dari alur kerja pemrosesan pinjaman yang lebih besar yang perlu multi-wilayah, platform perpesanan juga perlu mendukung persyaratan bisnis yang sama. Jika bisnis membutuhkan kemampuan untuk memulihkan dan mengembalikan ke keadaan sebelumnya dalam situasi hari hujan, platform pengiriman pesan atau streaming yang mendasarinya juga membutuhkan kemampuan untuk memiliki beberapa kemampuan snapshotting atau replay untuk menciptakan kembali keadaan sistem.

Platform integrasi yang Anda pilih harus memfasilitasi pemrosesan aplikasi pinjaman secara asinkron atau bertindak sebagai saluran penyimpanan dan penerusan untuk alur kerja pemrosesan media multi-langkah. Kekritisan proses bisnis akan menentukan kemampuan yang dibutuhkan dari platform pengiriman pesan atau streaming.

## Pertimbangkan

Saat mempertimbangkan arsitektur integrasi aplikasi utama di cloud, ada berbagai cara untuk menentukan persyaratan fungsional untuk masing-masing titik integrasi.

Berikut ini adalah beberapa kriteria yang perlu dipertimbangkan ketika memilih layanan integrasi aplikasi.

### Managed service and operation overhead

Pertimbangkan untuk pindah ke cloud untuk mengurangi biaya operasional dengan menstandarisasi layanan terkelola yang mengalihkan beban operasional. AWS Tingkat abstraksi yang lebih tinggi memungkinkan pengembang dan operator untuk fokus pada aktivitas nilai tambah unik mereka sendiri, bukan tugas yang tidak terdiferensiasi.

### Open source

Pertimbangkan standardisasi pada teknologi [open source](#). Open source dapat memungkinkan organisasi untuk menemukan keterampilan yang tepat—dan menghindari beberapa risiko seputar penguncian.

Membuat pilihan yang salah dalam ekosistem sumber terbuka dapat menyebabkan terkunci ke dalam abstraksi dan integrasi lokal. Selain itu, tanggung jawab untuk membuat komponen sumber terbuka yang berbeda bekerja bersama sering kali terletak pada organisasi yang membuat pilihan. Hal ini dapat menyebabkan organisasi menghabiskan waktu yang signifikan untuk mempertahankan integrasi sumber terbuka.

## Workload characteristics

Saat memilih layanan integrasi yang tepat, penting untuk memahami karakteristik pesan yang perlu dikirim antar aplikasi. Karakteristik utama seperti format pesan, ukuran, retensi dan prioritas dapat mendorong keputusan layanan integrasi.

Beberapa layanan integrasi lebih cocok untuk pesan berbasis teks kecil sedangkan beberapa dirancang untuk mendukung berbagai format seperti teks dan biner dan menawarkan ukuran pesan yang lebih besar. Kebutuhan untuk memiliki kemampuan replay juga bisa menjadi faktor penting di samping urutan pesan dalam beberapa skenario.

Misalnya, pemesanan pesan dapat diimplementasikan dengan menggunakan fungsionalitas FIFO yang ditawarkan oleh Amazon SNS dan Amazon SQS. Ada juga pertimbangan memiliki arsitektur berbasis pull atau push, seperti EventBridge atau Amazon SNS yang menjalankan fungsi Lambda secara asinkron.

Arsitektur berbasis tarik dapat menggunakan layanan seperti Amazon SQS atau Kinesis Data Streams, di mana pesan disimpan pada antrian atau aliran dan kemudian dapat diambil oleh sistem konsumsi. Layanan perpesanan seperti Amazon MQ menawarkan kemampuan di sekitar muatan pesan yang lebih besar dan memiliki retensi tak terbatas. Namun, mereka tidak menawarkan kemampuan replay.

## Rapid iteration and feature velocity

Jika fokus utama Anda adalah membangun dan mengulangi dengan cepat, layanan tanpa server mungkin memberikan nilai terbaik. Layanan tanpa server memungkinkan Anda membangun aplikasi tanpa mengelola infrastruktur. Mereka menyediakan fungsionalitas dan integrasi yang dikelola untuk mengurangi waktu yang dihabiskan untuk menulis kode pelat boiler.

Manfaat lain dari tanpa server saat menguji ide-ide baru adalah bahwa layanan ini menawarkan harga berbasis penggunaan. Kode Anda hanya berjalan saat layanan dipanggil, jadi eksperimen tidak memerlukan investasi di muka.

## Application portability

Banyak aplikasi menggunakan protokol tertentu - seperti Advanced Message Queuing Protocol (AMQP) atau MQ Telemetry Transport (MQTT) - untuk terhubung ke layanan pesan. Atau, mereka memiliki beberapa ketergantungan perpustakaan yang menggunakan protokol pesan tertentu. Contoh pustaka atau kerangka kerja tersebut termasuk Spring Boot, Celery, atau MassTransit

Anda mungkin ingin mempertahankan aplikasi tersebut untuk alasan yang berbeda. Dalam kasus ini, pilihan layanan integrasi Anda juga tergantung pada dukungan protokol yang diperlukan untuk memiliki portabilitas dengan aplikasi Anda.

### Automation portability

Anda mungkin perlu memiliki layanan yang menyediakan kompatibilitas dengan infrastruktur dan alat penyebaran Anda - dan menjalankan sistem integrasi yang sama dengan yang Anda host di tempat (seperti Apache ActiveMQ, RabbitMQ, dan Apache Kafka).

Layanan open source terkelola (seperti Amazon MQ dan Amazon MSK) memberikan manfaat cloud, sekaligus kompatibel dengan banyak alat penyebaran populer yang digunakan untuk penerapan lokal.

Jika refactoring aplikasi adalah pilihan, Anda bisa mendapatkan keuntungan dari menggunakan layanan tanpa server untuk menyediakan kemampuan ini secara asli, serta integrasi yang kaya dengan berbagai layanan. AWS

### Organization size and skills

Keterampilan organisasi Anda adalah faktor utama ketika memutuskan layanan integrasi yang tepat. Jika tim Anda terbiasa dengan produk yang dikelola sendiri dan memenuhi kebutuhan Anda, maka memiliki layanan yang dikelola untuk hal yang sama memberikan jalur yang paling tidak berdampak. Melakukannya, dapat membantu Anda menerapkan praktik terbaik untuk layanan dan fokus pada aktivitas nilai tambah.

## Pilih

Sekarang setelah Anda mengetahui kriteria yang akan Anda gunakan untuk mengevaluasi kebutuhan integrasi aplikasi Anda, Anda siap untuk memilih AWS layanan atau layanan mana yang tepat untuk beban kerja Anda di lingkungan Anda.

Jenis Layanan	Kapan Anda akan menggunakannya?	Untuk apa itu dioptimalkan?	Layanan terkait
Kapasitas	Gunakan saat Anda perlu memisahkan penerbit dan pelanggan dan	Dioptimalkan untuk komunikasi asinkron dan digabungkan secara longgar	<a href="#">Amazon EventBridge</a> <a href="#">Amazon SNS</a>

Jenis Layanan	Kapan Anda akan menggunakannya?	Untuk apa itu dioptimalkan?	Layanan terkait
Perpesanan	Gunakan saat Anda memerlukan pub/sub pesan untuk menyiarkan pesan ke beberapa penerima secara bersamaan, atau point-to-point mengirim pesan saat Anda membutuhkan komunikasi yang andal dan asinkron antar komponen.	Dioptimalkan untuk throughput tinggi, terukur, dan dapat diandalkan asinkron pub/sub dan pesan antar komponen terdistribusi. point-to-point	<a href="#">Amazon SNS</a> <a href="#">Amazon SQS</a> <a href="#">Amazon MQ</a>

Jenis Layanan	Kapan Anda akan menggunakannya?	Untuk apa itu dioptimalkan?	Layanan terkait
Streaming	Gunakan layanan streaming seperti Amazon Kinesis Data Streams dan Amazon Managed Streaming for Apache Kafka (MSK) dalam skenario yang melibatkan penanganan dan pemrosesan data streaming real-time.	Dioptimalkan untuk menelan, memproses, dan menganalisis volume besar data streaming real-time untuk kasus penggunaan yang memerlukan analitik real-time, pemantauan waktu nyata, eksplorasi data, dan aplikasi lain yang menuntut pemrosesan aliran data berkecepatan tinggi.	<a href="#">Amazon Kinesis Data Streams</a> <a href="#">Amazon MSK</a>

Jenis Layanan	Kapan Anda akan menggunakannya?	Untuk apa itu dioptimalkan?	Layanan terkait
Alur Kerja	Gunakan saat Anda perlu merancang, mengoordinasikan, dan mengelola alur kerja atau urutan tugas secara terorganisir dan terukur.	Dioptimalkan untuk kasus penggunaan seperti manajemen proses bisnis, orkestrasi aplikasi, otomatisasi pipa data, dan koordinasi layanan mikro. Alur kerja mengabstraksikan kompleksitas infrastruktur yang mendasarinya, memungkinkan Anda untuk fokus merancang dan mengelola alur kerja Anda secara efektif. Mereka mampu menangani dependensi dan pengurutan, memungkinkan paralelisme dan percabangan bersyarat sambil memberikan toleransi kesalahan, penanganan kesalahan, dan percobaan ulang untuk memastikan eksekusi alur kerja yang andal.	<a href="#">AWS Step Functions</a> <a href="#">Amazon MWAA</a>

Jenis Layanan	Kapan Anda akan menggunakannya?	Untuk apa itu dioptimalkan?	Layanan terkait
Penjadwalan	Gunakan penjadwalan saat Anda perlu mengotomatiskan tugas rutin seperti pemrosesan data, pencadangan, atau pemeriksaan kesehatan sistem. Tugas sering perlu dijalankan pada waktu atau interval tertentu, seperti setiap malam, jam, atau menit.	Dioptimalkan untuk tugas berbasis waktu yang andal dengan logika coba ulang bawaan. Cocokkan untuk alur kerja yang membutuhkan penjadwalan dan integrasi yang tepat dengan berbagai AWS layanan.	<a href="#">Amazon EventBridge</a>

## Gunakan

Anda sekarang harus memiliki pemahaman yang jelas tentang apa yang dilakukan setiap layanan integrasi AWS aplikasi — dan mana yang mungkin tepat untuk Anda. Untuk mengeksplorasi cara menggunakan dan mempelajari lebih lanjut tentang masing-masing layanan integrasi AWS aplikasi yang tersedia—kami telah menyediakan jalur untuk mengeksplorasi cara kerja masing-masing layanan. Bagian berikut menyediakan tautan ke dokumentasi mendalam, tutorial langsung, dan sumber daya untuk Anda mulai.

### Amazon SNS

- [Filter Pesan yang Diterbitkan ke Topik dengan Amazon SNS dan Amazon SQS](#)

Pelajari cara menggunakan fitur pemfilteran pesan Amazon SNS.

[Mulai menggunakan tutorial](#)

- [Amazon SNS - Pemecahan Masalah](#)

Pelajari cara melihat informasi konfigurasi, memantau proses, dan mengumpulkan data diagnostik tentang Amazon SNS.

### [Jelajahi panduan](#)

- Bangun game berbasis giliran dengan Amazon DynamoDB dan Amazon SNS

Pelajari cara membuat game multipemain berbasis giliran menggunakan Amazon DynamoDB dan Amazon SNS.

### [Mulai menggunakan tutorial](#)

- Membangun arsitektur berbasis acara

Pelajari cara membangun pub/sub implementasi sederhana menggunakan Amazon SNS sebagai layanan penerbitan kami dan Amazon SQS sebagai pelanggan.

### [Jelajahi panduan](#)

- Mengarsipkan dan memutar ulang pesan dengan Amazon SNS FIFO

Pelajari cara mengarsipkan dan memutar ulang pesan yang dipublikasikan ke Amazon SNS FIFO, yang dapat berguna dalam pemulihan kegagalan dan skenario replikasi status.

### [Baca posting blog](#)

## Amazon SQS

- Mulai menggunakan Amazon SQS

Panduan ini menunjukkan cara mengelola antrian dan pesan menggunakan konsol Amazon SQS.

### [Jelajahi panduan](#)

- Kirim Pemberitahuan Acara Fanout

Pelajari cara menerapkan skenario pesan fanout menggunakan Amazon SNS dan Amazon SQS.

### [Mulai menggunakan tutorial](#)

- Mengatur Layanan Mikro Berbasis Antrian

Pelajari cara mendesain dan menjalankan alur kerja tanpa server yang mengatur layanan mikro berbasis antrian pesan.

### [Mulai menggunakan tutorial](#)

- Kirim Pesan Antara Aplikasi Terdistribusi

Gunakan konsol Amazon SQS untuk membuat dan mengonfigurasi antrian pesan, mengirim pesan, menerima dan menghapus pesan itu, lalu menghapus antrean.

### [Mulai menggunakan tutorial](#)

## Amazon EventBridge

- Memulai dengan Amazon EventBridge

Dasarnya EventBridge adalah membuat aturan yang merutekan peristiwa ke target. Dalam panduan ini, Anda membuat aturan dasar.

### [Jelajahi panduan](#)

- Amazon EventBridge memulai tutorial

Tutorial ini akan membantu Anda menjelajahi fitur EventBridge dan cara menggunakannya.

### [Mulai menggunakan tutorial](#)

- Integrasi dengan yang lain Layanan AWS

Tutorial berikut menunjukkan cara mengintegrasikan EventBridge dengan yang lain Layanan AWS.

### [Mulai menggunakan tutorial](#)

- Membangun arsitektur berbasis peristiwa

Pelajari dasar-dasar desain berbasis peristiwa, cara memilih yang tepat Layanan AWS untuk pekerjaan itu, serta cara mengoptimalkan biaya dan kinerja.

### [Mulai menggunakan tutorial](#)

- Membangun aplikasi berbasis peristiwa dengan Amazon EventBridge

Pelajari cara membuat aplikasi berbasis peristiwa dengan menghubungkan beberapa aplikasi, termasuk aplikasi SaaS dan Layanan AWS, menggunakan bus acara tanpa server yang disediakan oleh Amazon. EventBridge

### [Mulai menggunakan tutorial](#)

- Konektor Kafka untuk Amazon EventBridge

Konektor ini memungkinkan Anda untuk mengubah catatan dari satu atau lebih topik Kafka menjadi acara, dan mengirim acara tersebut ke bus acara pilihan Anda.

### [Jelajahi panduan](#)

- Memperkenalkan target lintas akun untuk Amazon EventBridge Event Bus

Pelajari cara menggunakan Bus Acara untuk mengirim acara langsung ke target di tempat lain Akun AWS, seperti Amazon SQS, Amazon SNS, dan. AWS Lambda

### [Baca posting blog](#)

## Amazon MQ

- Buat broker pesan yang terhubung

Pelajari cara mengatur broker pesan Amazon MQ dan menghubungkan aplikasi Java tanpa menulis ulang kode Anda.

### [Mulai menggunakan tutorial](#)

- Migrasi ke broker pesan di cloud

Amazon MQ memudahkan migrasi ke broker pesan di cloud, seperti Apache ActiveMQ dan RabbitMQ.

### [Baca panduannya](#)

- Membuat dan menghubungkan ke broker RabbitMQ

Pelajari bagaimana Anda dapat menggunakan Konsol Manajemen AWS untuk membuat broker RabbitMQ dan melampirkan aplikasi Anda ke sana.

### [Mulai menggunakan tutorial](#)

- Lokakarya RabbitMQ

Lokakarya ini adalah kumpulan laboratorium yang mencakup berbagai aspek dan pola pengiriman pesan menggunakan RabbitMQ.

### [Memulai dengan Workshop](#)

- Memperkenalkan antrian kuorum di Amazon MQ untuk RabbitMQ

Temukan cara menggunakan antrian kuorum dengan RabbitMQ dapat memberikan ketersediaan dan keamanan data yang lebih tinggi.

### [Baca posting blog](#)

- Membuat dan menghubungkan ke broker ActiveMQ

Pelajari bagaimana Anda dapat menggunakan Konsol Manajemen AWS untuk membuat broker dasar.

### [Mulai menggunakan tutorial](#)

- Lokakarya ActiveMQ

Jelajahi konsep pesan seperti antrian, topik, dan fitur Amazon MQ seperti failover, jaringan broker.

### [Memulai dengan Workshop](#)

- Menyebarkan dan mempublikasikan ke broker Amazon MQ menggunakan tanpa server AWS

Kami memandu Anda menggunakan backend tanpa server dan broker Amazon MQ dalam satu langkah menggunakan SAM. AWS

### [Baca posting blog](#)

- Mengukur throughput Amazon MQ menggunakan benchmark Maven 2 dan AWS CDK

Pelajari cara mengevaluasi throughput Amazon MQ menggunakan plugin ActiveMQ Classic Maven Performance test.

### [Baca posting blog](#)

## Amazon Kinesis Data Streams

- Memulai dengan Amazon Kinesis Data Streams

Pelajari prinsip dasar aliran data Kinesis Data Streams dan langkah-langkah yang diperlukan untuk menempatkan dan mendapatkan data dari aliran data Kinesis.

### [Jelajahi panduan](#)

- Buat stream yang sangat tersedia dengan Amazon Kinesis Data Streams

Kami membandingkan dan membedakan berbagai strategi untuk membuat aliran data Kinesis yang sangat tersedia jika terjadi gangguan layanan, penundaan, atau pemadaman di Wilayah operasi utama.

### [Baca posting blog](#)

- Contoh Tutorial untuk Amazon Kinesis Data Streams

Tutorial ini dirancang untuk lebih membantu Anda dalam memahami konsep dan fungsionalitas Amazon Kinesis Data Streams.

### [Mulai menggunakan tutorial](#)

- Menggunakan AWS Lambda dengan Amazon Kinesis

Pelajari cara membuat fungsi Lambda untuk mengkonsumsi peristiwa dari aliran Kinesis.

### [Mulai menggunakan tutorial](#)

- Streaming Waktu Nyata dengan Amazon Kinesis

Jelajahi serangkaian latihan lab yang membantu pengguna membangun aplikasi analitik streaming AWS.

### [Mulai menggunakan tutorial](#)

## Amazon MSK

- Memulai menggunakan Amazon MSK

Tutorial ini menunjukkan contoh bagaimana Anda dapat membuat cluster MSK, memproduksi dan mengkonsumsi data, dan memantau kesehatan cluster Anda menggunakan metrik.

### [Mulai menggunakan tutorial](#)

- Memulai menggunakan kluster MSK Tanpa Server

Tutorial ini menunjukkan contoh bagaimana Anda dapat membuat kluster MSK Tanpa Server, membuat mesin klien yang dapat mengaksesnya, dan menggunakan klien untuk membuat topik di cluster dan menulis data ke topik tersebut.

### [Mulai menggunakan tutorial](#)

- Lab MSK Amazon

Laboratorium ini dapat dijalankan secara pribadi atau perusahaan Akun AWS atau akun yang disediakan oleh tim AWS Akun untuk acara yang menggunakan studio lokakarya.

### [Memulai dengan lab](#)

- Memperkenalkan broker Express untuk Amazon MSK untuk memberikan throughput tinggi dan penskalaan yang lebih cepat untuk cluster Kafka Anda

Pelajari bagaimana broker Express dapat menurunkan biaya, memberikan ketahanan yang lebih tinggi, dan menurunkan biaya operasional untuk beban kerja Kafka Anda.

### [Baca posting blog](#)

## AWS Step Functions

- Memulai dengan AWS Step Functions

Tutorial ini memandu Anda melalui pembuatan alur kerja dasar untuk memproses aplikasi kartu kredit.

### [Mulai menggunakan tutorial](#)

- Pengantar Step Functions

Kursus ini memperkenalkan komponen kunci dari Step Functions untuk membantu Anda memulai mengelola alur kerja dalam aplikasi.

### [Memulai dengan kursus](#)

- Pemrosesan data skala besar dengan AWS Step Functions

Pelajari bagaimana aplikasi pemrosesan data skala besar dapat dibangun menggunakan Step Functions.

### [Memulai dengan Workshop](#)

- Pola Desain untuk AWS Step Functions

Pelajari cara menerapkan pola desain di mesin status Step Functions Anda dan mengapa menggunakannya masing-masing.

#### [Memulai dengan kursus](#)

- Jadwalkan Alur Kerja Tanpa Server dengan dan AWS Step Functions Penjadwal Amazon EventBridge

Kami menunjukkan kepada Anda cara memanggil mesin status menggunakan EventBridge Scheduler berdasarkan jadwal yang Anda tentukan.

#### [Mulai menggunakan tutorial](#)

- AWS Step Functions Lokakarya

Pelajari cara menggunakan fitur utama AWS Step Functions melalui serangkaian modul interaktif.

#### [Memulai dengan Workshop](#)

- Berbagi AWS sumber daya secara aman di seluruh VPC dan batas akun dengan PrivateLink, VPC EventBridge Lattice, dan Step Functions

Pelajari cara berbagi AWS sumber daya, seperti EC2 instans dan layanan kontainer Amazon EKS, di seluruh VPC Amazon Akun AWS dan batasannya. AWS Step Functions

#### [Baca posting blog](#)

- Menyederhanakan pengalaman pengembang dengan variabel dan JSONata dalam AWS Step Functions

Jelajahi menggunakan variabel dan JSONata, kueri sumber terbuka dan bahasa transformasi, untuk menyederhanakan berbagi data di seluruh negara bagian dan mengurangi kompleksitas manipulasi data.

#### [Baca posting blog](#)

## Amazon MWAA

- Memulai dengan Amazon Managed Workflows for Apache Airflow

Panduan ini menjelaskan prasyarat dan AWS sumber daya yang diperlukan untuk memulai Amazon MWAA.

### [Jelajahi panduan](#)

- Mengkonfigurasi **aws-mwaa-local-runner** dalam pipa CD

Tutorial ini memandu Anda melalui proses pembuatan pipeline continuous delivery (CD) dalam GitHub menggunakan Amazon Managed Workflows untuk Apache Airflow untuk menguji kode Apache Airflow `aws-mwaa-local-runner` Anda secara lokal.

### [Mulai menggunakan tutorial](#)

- Membatasi akses pengguna Amazon MWAA ke subset DAGs

Kami menunjukkan bagaimana Anda dapat membatasi pengguna Amazon MWAA individu untuk hanya melihat dan berinteraksi dengan DAG tertentu atau satu set DAGs

### [Mulai menggunakan tutorial](#)

- Amazon MWAA untuk Lokakarya Analytics

Belajarlah untuk membangun dan mengatur data dan jaringan pipa ML yang mencakup banyak layanan yang disebutkan di atas, dan dengan itu Anda akan mendapatkan keakraban dan pemahaman yang lebih baik tentang kait dan operator yang tersedia sebagai bagian dari Aliran Udara untuk mengelola Anda. pipelines/workflows AWS

### [Memulai dengan Workshop](#)

## Jelajahi

Setelah Anda menentukan pendekatan mana yang paling sesuai dengan beban kerja Anda untuk lingkungan Anda, kami sarankan Anda meninjau sumber daya ini untuk membantu Anda mulai menerapkan pendekatan Anda. Anda dapat menemukan sumber daya khusus layanan di bagian sebelumnya, dan sumber daya arsitektur berbasis peristiwa umum di bagian berikut.

- Diagram arsitektur

Jelajahi diagram arsitektur referensi untuk membantu Anda membuat arsitektur yang sangat tersedia, aman, fleksibel, dan hemat biaya.

### [Jelajahi diagram arsitektur](#)

- Laporan Resmi

Jelajahi whitepaper untuk membantu Anda memulai, dan pelajari praktik terbaik seputar arsitektur berbasis acara.

### [Jelajahi whitepaper](#)

- Blog

Jelajahi blog untuk membantu Anda tetap up to date pada teknologi terbaru, dan memodernisasi aplikasi Anda.

### [Jelajahi blog](#)

## Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada panduan keputusan ini. Untuk pemberitahuan tentang pembaruan panduan ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
<a href="#">Panduan diperbarui</a>	Diperbarui untuk mencerminkan fitur baru dan banyak pembaruan editorial di seluruh.	April 16, 2025
<a href="#">Panduan diperbarui</a>	Tautan diperbarui.	Juni 25, 2024
<a href="#">Publikasi awal</a>	Panduan pertama kali diterbitkan.	31 Maret 2023

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.