# Làm phong phú và tùy chỉnh thông báo với Amazon EventBridge Pipes

bởi Chris McPeek vào08 THÁNG 4 NĂM 2025 trong [Amazon EventBridge](https://aws.amazon.com/blogs/compute/category/application-integration/amazon-eventbridge/) , [AWS Lambda](https://aws.amazon.com/blogs/compute/category/compute/aws-lambda/) , [Serverless](https://aws.amazon.com/blogs/compute/category/serverless/) [Liên kết cố định](https://aws.amazon.com/blogs/compute/enriching-and-customizing-notifications-with-amazon-eventbridge-pipes/) [Chia sẻ](https://aws.amazon.com/vi/blogs/compute/enriching-and-customizing-notifications-with-amazon-eventbridge-pipes/#)

*Bài đăng trên blog này được viết bởi Elie Elmalem, Kiến trúc sư giải pháp quy mô cộng sự*

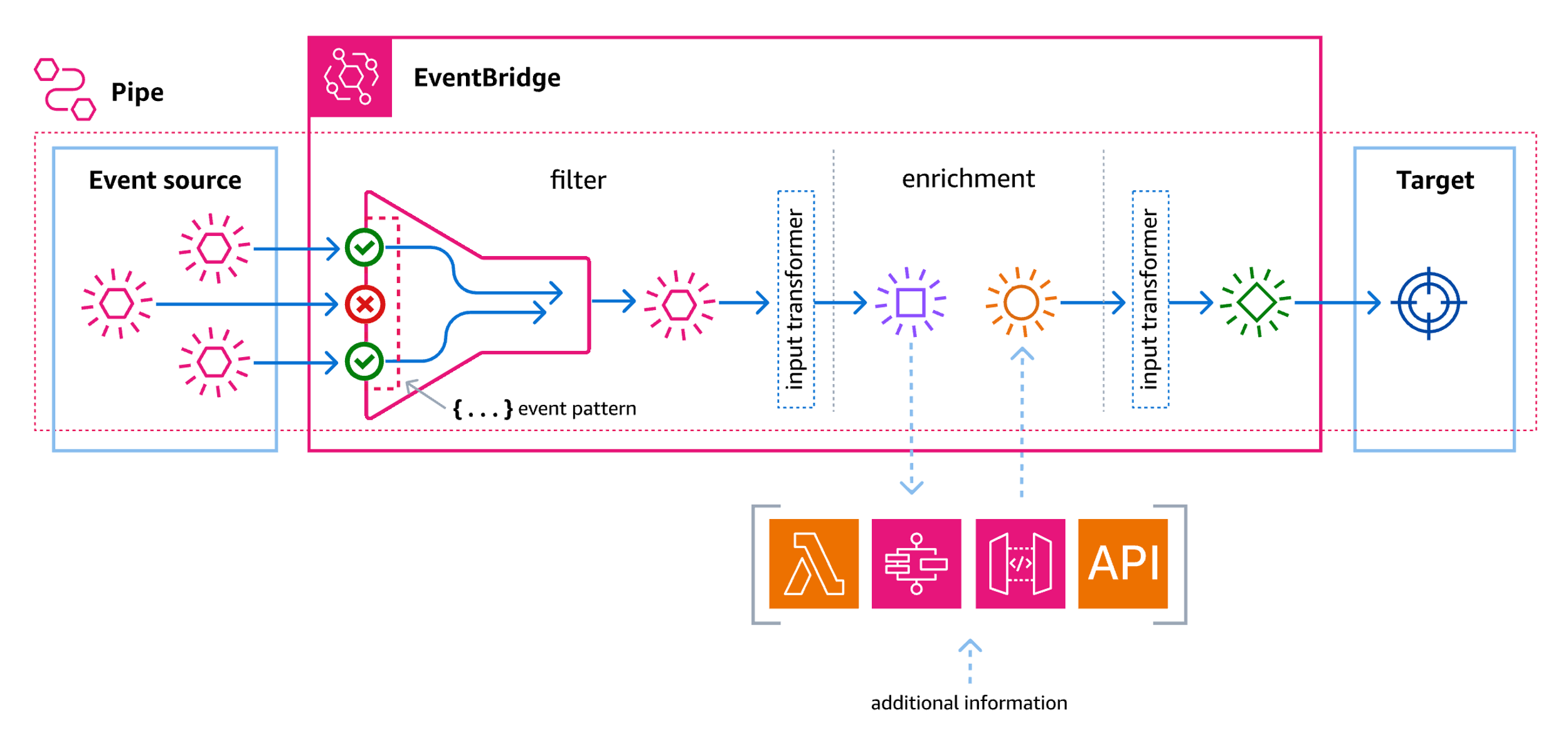
Khi triển khai kiến ​​trúc hướng sự kiện, khách hàng thường cần làm phong phú thêm các sự kiện đến của họ bằng thông tin bổ sung để làm cho chúng có giá trị hơn đối với người tiêu dùng hạ nguồn. Theo truyền thống, khách hàng sử dụng [Amazon EventBridge](https://aws.amazon.com/eventbridge/) sẽ thực hiện điều này bằng cách viết các hàm [AWS Lambda](https://docs.aws.amazon.com/lambda/) để tăng cường sự kiện của họ bằng dữ liệu bổ sung. Tuy nhiên, cách tiếp cận này đòi hỏi phải viết và duy trì mã tùy chỉnh, làm tăng thêm tính phức tạp cho đường ống xử lý sự kiện của họ.

[Amazon EventBridge Pipes](https://aws.amazon.com/fr/eventbridge/pipes/) đơn giản hóa quy trình này bằng cách cung cấp dịch vụ được quản lý, hợp lý hóa để làm giàu sự kiện mà không cần phải viết và quản lý các hàm Lambda tùy chỉnh. Bài đăng trên blog này trình bày cách bạn có thể sử dụng các khả năng làm giàu dữ liệu tích hợp của EventBridge Pipes để nâng cao sự kiện của bạn một cách năng động với ngữ cảnh bổ sung và thông tin chi tiết cụ thể của khách hàng, giúp xử lý sự kiện hiệu quả hơn và dễ bảo trì hơn.

## **Ống Amazon EventBridge**

Amazon EventBridge tạo ra kết nối trực tiếp giữa [các nguồn](https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-pipes-event-source.html) và [đích](https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-pipes-event-target.html) . Sử dụng bus EventBridge giúp bạn định tuyến và phân phối sự kiện đến các dịch vụ theo mô hình pub/sub. Mặt khác, EventBridge Pipes giúp bạn với các mô hình tích hợp dịch vụ điểm-đến-điểm. Điểm khác biệt của nó so với mô hình event bus/rule truyền thống là hỗ trợ chuyển đổi dữ liệu và làm giàu dữ liệu.

Khi định nghĩa EventBridge Pipes, bạn chỉ định nguồn và đích của pipe. Pipes hỗ trợ nhiều nguồn và đích khác nhau. Giữa nguồn và đích, EventBridge Pipes hỗ trợ [lọc](https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-pipes-event-filtering.html) và [làm giàu](https://docs.aws.amazon.com/eventbridge/latest/userguide/pipes-enrichment.html) . Lọc cho phép bạn chọn và xử lý một tập hợp con sự kiện mục tiêu. Làm giàu cho phép bạn tăng cường dữ liệu bằng cách thêm thông tin bị thiếu trước khi gửi đến đích. Ví dụ: nếu sự kiện thiếu thông tin cần thiết, nó đảm bảo đích có thể sử dụng sự kiện đó một cách phù hợp. Làm giàu dữ liệu có thể rất mạnh mẽ vì nó giúp tăng cường sự kiện chung và chuyển đổi sự kiện đó. EventBridge Pipes hỗ trợ làm giàu bằng cách sử dụng các hàm Lambda, [AWS Step Functions](https://aws.amazon.com/step-functions/?nc1=h_ls) , [Amazon API Gateway](https://aws.amazon.com/api-gateway/) và [các đích API EventBridge](https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-api-destinations.html) . Bạn có thể tìm hiểu thêm thông tin chi tiết về các khái niệm này trong tài liệu [về khái niệm Amazon EventBridge Pipes](https://docs.aws.amazon.com/eventbridge/latest/userguide/pipes-concepts.html) .



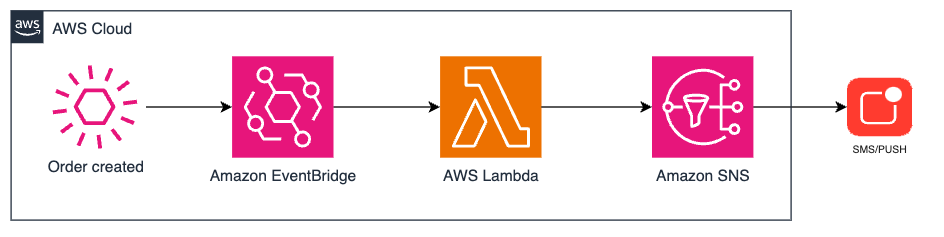
Hình 1: Biểu diễn của EventBridge Pipes hiển thị các bước lọc và làm giàu

Bài đăng trên blog này sẽ sử dụng bước làm giàu của đường ống để tạo thông báo tùy chỉnh.

## **Tổng quan**

Để minh họa chức năng, bài đăng này sử dụng trường hợp sử dụng từ một nhà bán lẻ quần áo. Các doanh nghiệp như nhà bán lẻ này muốn giữ chân khách hàng trung thành của mình. Họ thường dựa vào các email quảng cáo hàng loạt thiếu tính cá nhân hóa. Trong trường hợp sử dụng này, nhà bán lẻ muốn gửi mã khuyến mại có mục tiêu. Ngay khi đơn hàng thứ 10 được đặt, mã sẽ được gửi qua email hoặc SMS đến khách hàng của họ.

Nếu không có EventBridge Pipes, điều này sẽ được triển khai bằng EventBridge để phản hồi sự kiện đơn hàng. Tất cả các sự kiện được gửi đến một hàm Lambda tùy chỉnh để xử lý. Nếu đơn hàng đáp ứng đúng các điều kiện, hàm Lambda sẽ gửi thông báo có mã giảm giá cho khách hàng bằng [Amazon Simple Notification Service (Amazon SNS)](https://docs.aws.amazon.com/sns/latest/dg/welcome.html) .

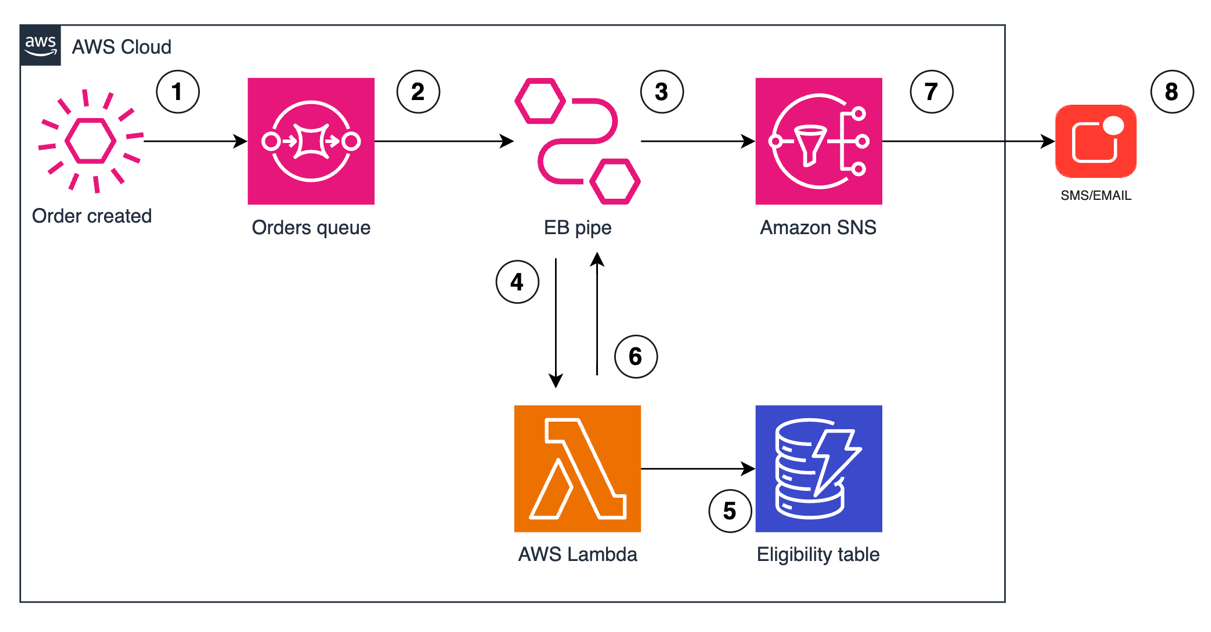


Hình 2: Phương pháp truyền thống sử dụng EventBridge

Trong khi kiến ​​trúc này hoạt động, nó yêu cầu bạn phải duy trì mã tích hợp cũng như logic làm giàu dữ liệu trong hàm Lambda vì hàm này cần trích xuất thông tin cần thiết từ các sự kiện và quản lý định tuyến đến SNS. Khi nhiều dịch vụ vi mô tuân theo cùng một mẫu, mã sẽ trở nên phức tạp hơn. Điều này có thể dẫn đến thời gian thực hiện lâu hơn cùng với chi phí cao hơn và nỗ lực bảo trì lớn hơn.

## **Đơn giản hóa bằng cách sử dụng Amazon EventBridge Pipes**

Amazon EventBridge Pipes có thể được sử dụng để đơn giản hóa việc triển khai trước đó bằng cách xử lý việc làm giàu và tích hợp giữa các dịch vụ. Amazon EventBridge Pipes đảm nhiệm việc gửi sự kiện đến bước làm giàu đã cấu hình của bạn và sau đó định tuyến sự kiện đã làm giàu đến mục tiêu. Nếu phương pháp được chọn là hàm Lambda, nó sẽ để mã hàm chỉ tập trung vào logic làm giàu. Nó loại bỏ nhu cầu về mã để trích xuất các trường cần thiết từ sự kiện và gửi thông báo.



Hình 3: Kiến trúc giải pháp sử dụng EventBridge Pipes

Khi sự kiện đi vào đường ống, bước làm giàu sẽ kích hoạt hàm Lambda, hàm này sẽ kiểm tra tính đủ điều kiện và trả về tin nhắn để định tuyến đến SNS. Nếu khách hàng không đủ điều kiện để được giảm giá, nó sẽ trả về tin nhắn xác nhận đơn hàng với dữ liệu được lấy từ sự kiện đơn hàng ban đầu. Nếu khách hàng đủ điều kiện để được giảm giá, tin nhắn cũng chứa mã giảm giá.

Đây là kiến ​​trúc cho luồng được cập nhật:

1. Một khách hàng đặt hàng một mặt hàng mới. Đơn hàng được gửi đến hàng đợi Đơn hàng [của Dịch vụ hàng đợi đơn giản (SQS)](https://aws.amazon.com/sqs/) .
2. Tin nhắn mới trên hàng đợi Đơn hàng sẽ kích hoạt EventBridge Pipes.
3. Đường ống kích hoạt chức năng AWS Lambda để làm giàu dữ liệu.
4. Các hàm này kiểm tra xem khách hàng có đủ điều kiện để nhận mã giảm giá theo bảng [Amazon DynamoDB](https://aws.amazon.com/dynamodb/) hay không. Bảng này chứa số lần mỗi khách hàng đã đặt hàng.
5. Hàm Lambda trả về tin nhắn tùy chỉnh sẽ được gửi tới khách hàng, có hoặc không có mã giảm giá.
6. Tin nhắn được định tuyến đến một chủ đề SNS bởi EventBridge Pipe
7. Khách hàng nhận được thông báo qua phương thức đăng ký ưa thích của mình.

## **Xây dựng luồng cập nhật**

Để xây dựng luồng được cập nhật, tôi đã chọn sử dụng [AWS Cloud Development Kit (CDK)](https://aws.amazon.com/cdk/) trong Python. Bạn có thể sử dụng mã được cung cấp ở đây để triển khai vào tài khoản của mình. Mã cũng có thể được tìm thấy trên [GitHub](https://github.com/aws-samples/sample-enriching-and-customizing-notifications-with-amazon-eventBridge-pipes) .

Lưu ý : Mã mẫu này chỉ nhằm mục đích thử nghiệm và không nhằm mục đích sử dụng trong tài khoản sản xuất.

Đối với giải pháp này, bạn cần có những điều kiện tiên quyết sau:

1. Giao [diện dòng lệnh AWS (CLI)](https://aws.amazon.com/cli/) được cài đặt và [cấu hình để sử dụng](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html) .
2. Vai trò hoặc người dùng [Quản lý danh tính và truy cập (IAM)](https://aws.amazon.com/iam/) có đủ quyền để tạo chính sách IAM, bảng DynamoDB, hàng đợi SQS, chủ đề SNS, Hàm Lambda và Đường ống EventBridge.
3. [AWS CDK](https://docs.aws.amazon.com/cdk/v2/guide/getting_started.html)
4. [Python phiên bản 3.9](https://www.python.org/downloads/release/python-390/) trở lên, có [pip và virtual virtualenv](https://docs.aws.amazon.com/cdk/v2/guide/work-with-cdk-python.html#python-prerequisites) .

Khi đã đáp ứng được các điều kiện tiên quyết, hãy thiết lập một dự án Python CDK mới trong một thư mục trống:

mkdir blog\_code

cd blog\_code

cdk init app –-language python

Sau đó, kích hoạt môi trường ảo và cài đặt các phần phụ thuộc của CDK:

source .venv/bin/activate

python -m pip install -r requirements.txt

Lệnh cdk init tạo một thư mục blog\_code. Kho lưu trữ GitHub chứa mã cho tệp blog\_code\_stack.py bên trong thư mục blog\_code.

Sau đó, trong thư mục blog\_code, tạo một thư mục mới có tên lambda. Trong thư mục mới này, tạo một tệp có tên index.py. Tệp này sẽ chứa mã cho hàm lambda làm giàu. Một lần nữa, mã này có thể được tìm thấy trong kho lưu trữ GitHub. Sau đây là một phần của mã Lambda:

def lambda\_handler(event, context):

message = json.loads(event[0]['body'])

id = message['id']

order\_content = message['order\_content']

nmb\_orders = get\_number\_of\_orders(id)

# Calculate orders left

orders\_left = MAX\_ORDERS - nmb\_orders

# Update the DynamoDB table with the new number of orders

if nmb\_orders == MAX\_ORDERS:

update\_table(id, 0)

else:

update\_table(id, nmb\_orders)

if orders\_left == 0:

return [f"Thank you for your order of {order\_content}. You have earned a 10% discount code on your next order: XA5GT2SF"]

else:

# Return the confirmation message

return [f"Thank you for your order of {order\_content}. This is your confirmation message! Only {orders\_left} orders left until a 10% discount!"]

Trăn

Hàm Lambda hoạt động theo cách sau:

1. Nó nhận một sự kiện từ EventBridge Pipe bao gồm đơn hàng và ID của người dùng đã thực hiện đơn hàng
2. Lệnh này sẽ lấy số lượng đơn hàng mà người dùng đã đặt bằng cách gọi lệnh *GetItem* trên bảng DynamoDB.
3. Công cụ này tính toán số lượng đơn hàng còn lại trước khi người dùng nhận được mã giảm giá.
4. Nó cập nhật bảng DynamoDB với số lượng đơn hàng mới để tính cho đơn hàng vừa được đặt.
5. Nếu người dùng đã đặt đúng số lượng đơn hàng, hệ thống sẽ trả về tin nhắn xác nhận có mã giảm giá. Nếu không, hệ thống sẽ thông báo cho người dùng số lượng đơn hàng vẫn cần phải đặt để được giảm giá.

Bây giờ, triển khai ngăn xếp CDK vào tài khoản của bạn. Đảm bảo rằng bạn đang ở trong thư mục gốc của dự án:

cdk bootstrap

cdk deploy

Sau khi ngăn xếp triển khai xong, bạn sẽ thấy đường ống EventBridge hiển thị trên bảng điều khiển bằng cách vào [trang dịch vụ EventBridge](https://console.aws.amazon.com/events/home) và nhấp vào Đường ống ở bảng điều khiển bên trái.

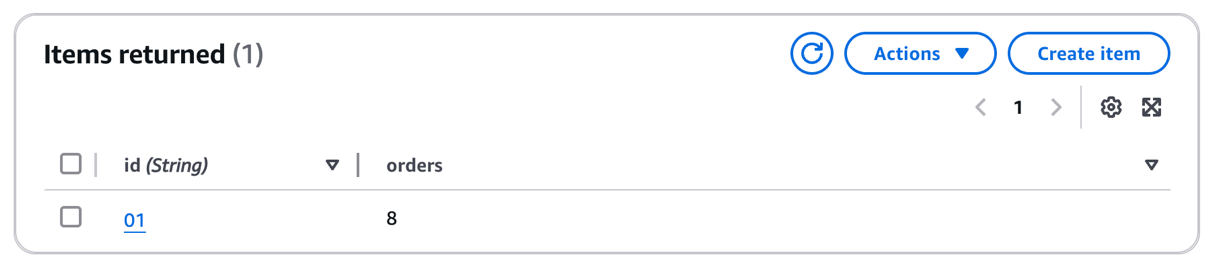
## **Kiểm tra giải pháp**

Để kiểm tra giải pháp, trước tiên bạn phải thiết lập đăng ký chủ đề SNS để nhận thông báo. Nên thiết lập thông báo qua email để đơn giản và mục đích thử nghiệm. Để thực hiện, hãy làm theo hướng dẫn trong [tài liệu Amazon SNS](https://docs.aws.amazon.com/sns/latest/dg/sns-email-notifications.html#create-subscribe-endpoint-to-topic-console) cho chủ đề có tên TargetTopic. Khi đăng ký được thiết lập, đừng quên kiểm tra hộp thư đến email của bạn và xác nhận đăng ký.

Sau khi thiết lập thông báo, hãy truy cập [trang bảng điều khiển DynamoDB](https://console.aws.amazon.com/dynamodbv2/home) . Bạn cần thêm thủ công một mục vào bảng đủ điều kiện để mô phỏng môi trường thực tế:

1. Nhấp vào Bảng ở bảng bên trái
2. Chọn bảng EligibilityTable .
3. Nhấp vào Khám phá các mục bảng sau đó Tạo mục
4. Nhập một id có giá trị là 01 .
5. Nhấp vào Thêm thuộc tính mới và chọn Chuỗi .
6. Trong tên thuộc tính, nhập orders và trong giá trị, nhập 8 .
7. Nhấp vào Tạo mục .

Bảng Items returns sẽ trông như sau. Điều này giả định rằng khách hàng đã đặt 8 đơn hàng.



Hình 4: Bảng các mục được trả về sau khi thêm một mục mới

Bây giờ, hãy truy cập [trang bảng điều khiển SQS](https://console.aws.amazon.com/sqs/home) . Bạn sẽ cần gửi tin nhắn đến hàng đợi để mô phỏng các đơn hàng mới được đặt.

1. Nhấp vào hàng đợi có tên là SourceQueue .
2. Nhấp vào Gửi và nhận tin nhắn .
3. Trong nội dung tin nhắn, dán tin nhắn sau vào và nhấp vào Gửi tin nhắn :

{

"order\_content": "large shirt",

"id": "01",

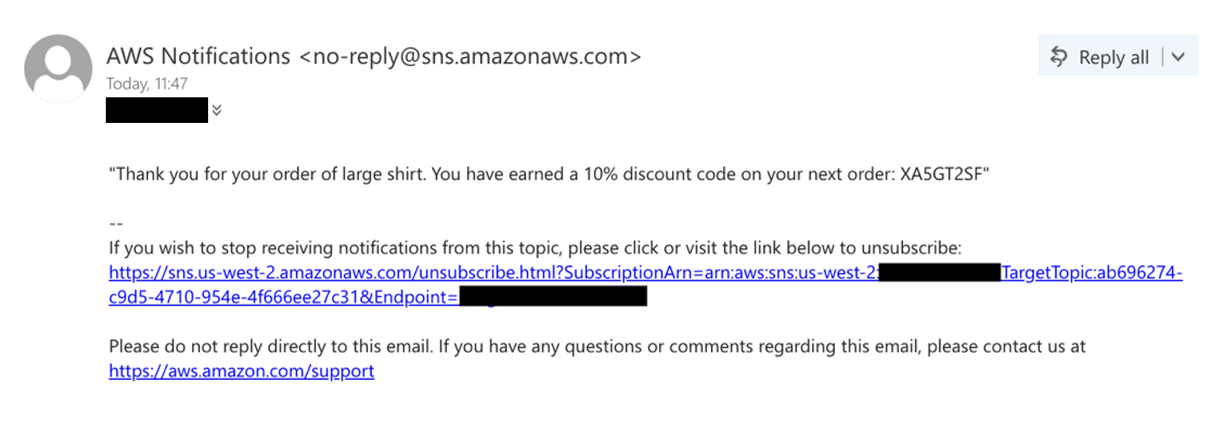
"username": "johndoe01",

"transaction\_time": "10:04:00"

}

JSON

Sau vài phút, bạn sẽ nhận được email xác nhận đơn hàng của mình vì tin nhắn đặt hàng của bạn được coi là đơn hàng thứ 9. Gửi lại tin nhắn để đặt đơn hàng thứ 10 và bạn sẽ nhận được mã giảm giá!



Hình 5: Email nhận được có mã giảm giá

## **Dọn dẹp**

Để xóa tài nguyên trong tài khoản của bạn, hãy chạy lệnh sau trong thư mục gốc của dự án:

cdk destroy

## **Phần kết luận**

Bài đăng trên blog này cho thấy Amazon EventBridge Pipes và tính năng làm giàu của nó có thể giúp bạn tạo thông báo tùy chỉnh như thế nào. Đầu tiên, bài đăng thảo luận về cách triển khai bằng EventBridge và sau đó trình bày cách triển khai đơn giản bằng EventBridge Pipes.

Để biết thêm thông tin về các mẫu phổ biến cho EventBridge Pipes, bạn có thể xem Triển khai [các mẫu kiến ​​trúc với Amazon EventBridge Pipes](https://aws.amazon.com/blogs/compute/implementing-architectural-patterns-with-amazon-eventbridge-pipes/) .

Để biết thêm tài nguyên học tập không có máy chủ, hãy truy cập [Serverless Land](https://serverlessland.com/) . Để tìm thêm các mẫu, hãy truy cập trực tiếp vào [Bộ sưu tập mẫu không có máy chủ](https://serverlessland.com/patterns) .

THẺ: [đóng góp](https://aws.amazon.com/blogs/compute/tag/contributed/) , [không có máy chủ](https://aws.amazon.com/blogs/compute/tag/serverless/)