Điều phối xử lý tài liệu với AWS AppSync Events và Amazon Bedrock

**📖 Bài viết gốc:** <https://aws.amazon.com/vi/blogs/compute/orchestrating-document-processing-with-aws-appsync-events-and-amazon-bedrock/>

**👤 Tác giả:** Mehdi Amrane  
**📅 Ngày xuất bản:** 09 tháng 7 năm 2025  
**🌐 Nguồn:** AWS Blog  
**👨‍💻 Người dịch:** Nguyễn Quang Bình  
**📅 Ngày dịch:** 08/07/2025  
**⏱️ Thời gian đọc:** 15 phút

──────────────────────────────────────────────────

## 📋 Tóm tắt

Nhiều tổ chức triển khai các đường ống xử lý tài liệu thông minh để trích xuất những hiểu biết có ý nghĩa từ khối lượng ngày càng tăng của nội dung không có cấu trúc (như yêu cầu bảo hiểm, đơn xin vay và nhiều hơn nữa). Theo truyền thống, những đường ống này đòi hỏi nỗ lực kỹ thuật đáng kể, vì việc ...

**🎯 Đối tượng đọc:** Developers, Solution Architects, DevOps Engineers  
**📊 Độ khó:** Intermediate  
**🏷️ Tags:** AWS, Cloud Computing, Architecture

──────────────────────────────────────────────────

## 📚 Mục lục

• Giới thiệu  
• Kiến trúc giải pháp  
• Triển khai  
• Kết luận  
• Glossary - Thuật ngữ  
• Tài liệu tham khảo

──────────────────────────────────────────────────

Nhiều tổ chức triển khai các đường ống xử lý tài liệu thông minh để trích xuất những hiểu biết có ý nghĩa từ khối lượng ngày càng tăng của nội dung không có cấu trúc (như yêu cầu bảo hiểm, đơn xin vay và nhiều hơn nữa). Theo truyền thống, những đường ống này đòi hỏi nỗ lực kỹ thuật đáng kể, vì việc triển khai thường liên quan đến việc sử dụng một số mô hình học máy (ML) và điều phối các quy trình làm việc phức tạp.

Khi các tổ chức tích hợp những đường ống này vào các ứng dụng hướng tới khách hàng (như ứng dụng web cho khách hàng tải lên tài liệu như yêu cầu bảo hiểm, tài liệu phê duyệt vay và nhiều hơn nữa), họ đặt mục tiêu cung cấp hiểu biết theo thời gian thực để tăng trải nghiệm khách hàng cuối. Những tổ chức này cũng nhằm chạy và mở rộng những khối lượng công việc này với chi phí vận hành tối thiểu và tối ưu hóa chi phí. Ngoài ra, những tổ chức này yêu cầu triển khai các thực hành bảo mật phổ biến như quản lý danh tính và truy cập, để đảm bảo rằng chỉ những người dùng được ủy quyền và xác thực mới được phép thực hiện các hành động cụ thể hoặc truy cập các tài nguyên cụ thể.

Trong bài viết này, chúng tôi cho bạn thấy một giải pháp để đơn giản hóa việc tạo ra một đường ống xử lý tài liệu thông minh, với một ứng dụng web cho khách hàng tải lên các tệp của họ (tài liệu và hình ảnh) và rút ra hiểu biết từ nó (tóm tắt, trích xuất trường và phân loại). Giải pháp chủ yếu sử dụng các công nghệ serverless, nó bao gồm một web socket để nhận hiểu biết theo thời gian thực và cung cấp một số lợi ích, như tự động mở rộng, tính khả dụng cao tích hợp sẵn, và mô hình thanh toán theo sử dụng để tối ưu hóa chi phí. Giải pháp cũng bao gồm một lớp xác thực và một lớp ủy quyền để quản lý danh tính và quyền hạn.

**Tổng quan Giải pháp**

Trong bài viết này, chúng tôi cung cấp một tổng quan hoạt động của giải pháp, và sau đó mô tả cách thiết lập nó với các dịch vụ sau:

Amazon Bedrock và Amazon Bedrock Data Automation để tóm tắt nội dung của các tệp được tải lên (tài liệu hoặc hình ảnh) và tạo ra hiểu biết từ nó

AWS Step Functions và AWS Lambda để điều phối các hoạt động tóm tắt và trích xuất, sử dụng Amazon Bedrock và Amazon Bedrock Data Automation

AWS AppSync Events để tạo một websocket serverless để ứng dụng web nhận được hiểu biết tóm tắt và trích xuất theo thời gian thực

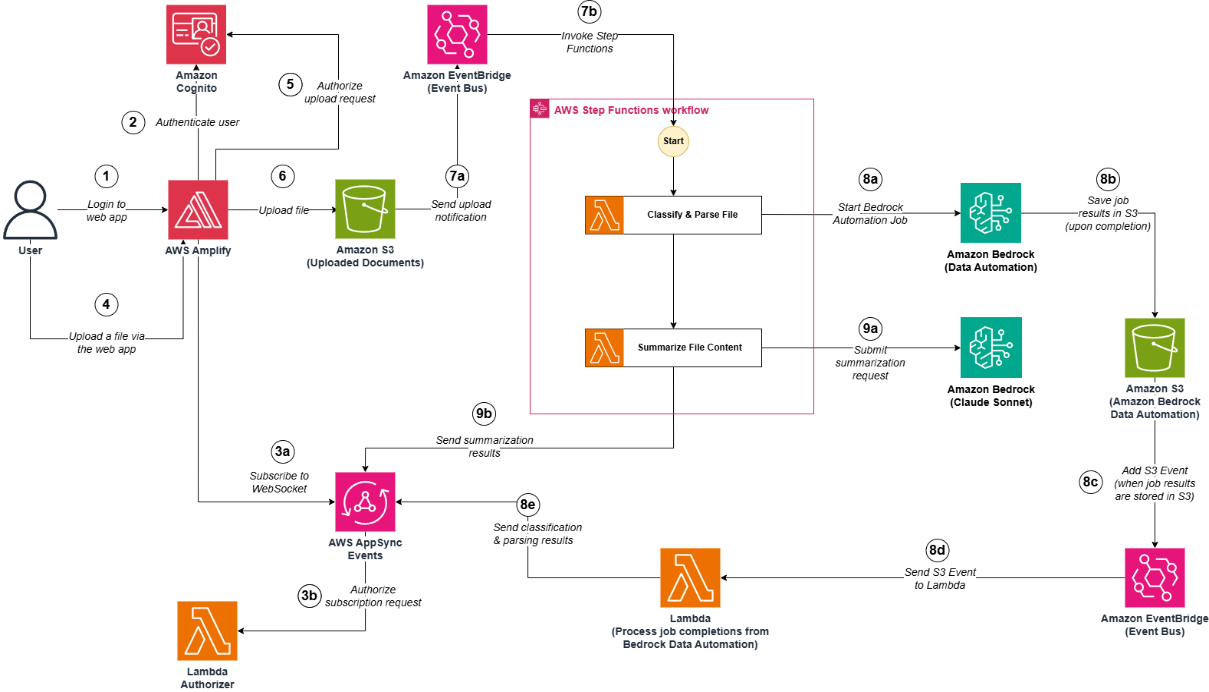
AWS Amplify để tạo và triển khai ứng dụng web

Amazon EventBridge để kích hoạt quy trình làm việc điều phối (sử dụng AWS Step Functions và AWS Lambda) khi tải lên một tệp mới

Amazon Cognito để triển khai một nền tảng danh tính (thư mục người dùng và quản lý ủy quyền) cho ứng dụng web

Amazon Simple Storage Service (Amazon S3) để lưu trữ các tệp được tải lên (để được xử lý bởi đường ống xử lý) và các tài sản liên quan đến ứng dụng web

Kiến trúc giải pháp được minh họa trong sơ đồ sau:



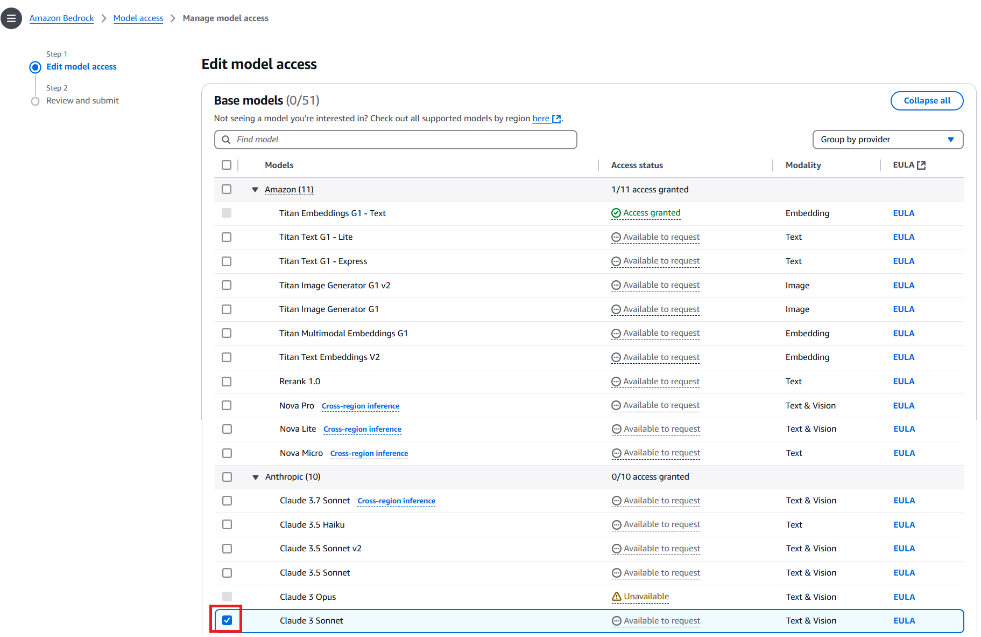
Bước 1: Người dùng xác thực vào ứng dụng web (được lưu trữ trong AWS Amplify).  
Bước 2: Amazon Cognito xác thực chi tiết xác thực. Sau đó, người dùng hiện đã đăng nhập vào ứng dụng web.  
Bước 3a và 3b:  
• Bước 3a: Ứng dụng web (AWS Amplify) đăng ký vào một AWS AppSync Events web socket.  
• Bước 3b: AWS AppSync Events web socket gọi một AWS Lambda authorizer để xác nhận rằng người dùng được ủy quyền đăng ký vào web socket.  
Bước 4: Người dùng tải lên một tệp (tài liệu hoặc hình ảnh) sử dụng ứng dụng web.  
Bước 5: Ứng dụng web (được lưu trữ trong AWS Amplify) gọi Amazon Cognito (identity pool) để xác nhận rằng người dùng được ủy quyền tải lên một tệp.  
Bước 6: Tệp được tải lên trong một Amazon S3 bucket.  
Bước 7a và 7b: Khi nhận được một sự kiện tải lên Amazon S3 (thông báo rằng tệp đã được tải lên trong Amazon S3 bucket) trong bus Amazon Event Bridge mặc định, một quy tắc bus Amazon Event Bridge kích hoạt việc thực thi một AWS Step Functions state machine để bắt đầu quy trình làm việc điều phối.  
Bước 8 (Bước để trích xuất trường từ một tệp và phân loại nó):  
• Bước 8a: AWS Lambda function đầu tiên bắt đầu một công việc Amazon Bedrock Automation mới (công việc này trích xuất các trường cụ thể từ tệp được tải lên và phân loại nó)  
• Bước 8b: Một khi công việc hoàn thành, kết quả được lưu trữ trong một Amazon S3 bucket.  
• Bước 8c và 8d: Khi nhận được một sự kiện Amazon S3 (thông báo rằng kết quả đã được lưu trữ trong Amazon S3 bucket) trong Amazon Event Bridge mặc định, một quy tắc bus Amazon Event Bridge kích hoạt việc thực thi một AWS Lambda function  
• Bước 8e: Một AWS Lambda function xuất bản kết quả lên web socket.  
Bước 9a và 9b: AWS Lambda function thứ hai gửi một prompt đến một mô hình nền tảng Amazon Bedrock (Sonnet 3), để yêu cầu một tóm tắt streaming của tệp được tải lên. AWS Lambda function xuất bản dữ liệu streaming lên web socket.

Sau Bước 8e và Bước 9b, người dùng giờ có thể tham khảo kết quả tóm tắt và hiểu biết trích xuất của tệp được tải lên trong ứng dụng web.

**Điều kiện tiên quyết**

Để theo dõi và thiết lập giải pháp này, bạn phải có những điều sau:

* Một tài khoản AWS
* Một thiết bị có quyền truy cập vào tài khoản AWS của bạn với những điều sau:
* Python 3.12 được cài đặt (bao gồm pip)
* Node.js 20.12.0 được cài đặt
* Kích hoạt Model Access đến mô hình Claude 3 Sonnet trong Amazon Bedrock



Lưu ý: Triển khai giải pháp này sẽ phát sinh chi phí. Xem lại trang giá của mỗi dịch vụ AWS được sử dụng trong bài viết này để biết chi tiết về chi phí. Chi phí chạy giải pháp này sẽ chủ yếu phụ thuộc vào:  
• Số lượng tài liệu (và kích thước của mỗi tài liệu)  
• Số lượng người dùng hoạt động

**Thiết lập Amazon Bedrock Data Automation**

Trong phần này, chúng tôi thiết lập một dự án Amazon Bedrock Data Automation và một blueprint Amazon Bedrock.

Một dự án chứa một danh sách các blueprint, và mỗi blueprint định nghĩa các trường để trích xuất từ các loại tệp khác nhau (như tài liệu hoặc hình ảnh). Trong bài viết này, chúng tôi định nghĩa một blueprint cho bằng lái xe.

Hoàn thành các bước sau để tạo một dự án Amazon Bedrock Data Automation và một blueprint bằng lái xe:

1. Clone GitHub repository  
*git clone https://github.com/aws-samples/sample-create-idp-with-appsyncevents-and-amazonbedrock.git*

2. Đi đến thư mục sample-create-idp-with-appsyncevents-and-amazonbedrock  
*cd sample-create-idp-with-appsyncevents-and-amazonbedrock*

3. Khởi tạo môi trường (làm cho các tệp shell script, từ GitHub repository, sẵn sàng để sử dụng)  
*chmod +x ./init-env.sh && source ./init-env.sh*

4. Chạy script setup-bda-project.sh để tạo một dự án Amazon Bedrock Data Automation và một blueprint bằng lái xe mẫu:  
*./setup-bda-project.sh*

**Tạo web socket và backend điều phối**

Trong phần này, chúng tôi tạo các tài nguyên sau:

Một thư mục người dùng cho xác thực và ủy quyền web, được tạo với một Amazon Cognito user pool. Một Amazon Cognito identity pool cũng được tạo để xác thực rằng người dùng được ủy quyền tải lên tệp qua ứng dụng web.

Một web socket sử dụng AWS AppSync Events. Điều này cho phép ứng dụng web của chúng tôi nhận cập nhật thời gian thực cho kết quả tóm tắt và trích xuất. Một lớp ủy quyền cũng được tạo để bảo vệ web socket khỏi người dùng không được ủy quyền. Điều này được triển khai với một Lambda authorizer function để xác thực rằng các yêu cầu đến bao gồm chi tiết ủy quyền hợp lệ.

Một state machine sử dụng AWS Step Functions và AWS Lambda để điều phối các hoạt động tóm tắt và trích xuất từ nội dung không có cấu trúc

Amazon S3 buckets để lưu trữ tệp cho xử lý tài liệu, và các tệp mã cho AWS Lambda functions

Hoàn thành các bước sau để tạo web socket và backend điều phối của giải pháp, sử dụng AWS CloudFormation templates:

1. Tạo Amazon S3 buckets được sử dụng bởi giải pháp bằng cách chạy script sau. Những buckets này sẽ lưu trữ các tệp được tải lên bởi người dùng và các tệp mã của AWS Lambda functions được sử dụng trong giải pháp này.  
*cd $CURRENT\_DIR/s3; ./create-s3-buckets.sh*

2. Tạo Amazon Cognito user pool và identity pool bằng cách chạy script create-cognito-userpool.sh:  
*cd $CURRENT\_DIR/cognito; ./create-cognito-userpool.sh*

3. Tạo AWS AppSync Events web socket bằng cách chạy script sau:  
*cd $CURRENT\_DIR/appsync/; ./create-appsync-api.sh*

4. Tạo AWS Step Functions state machine (bao gồm AWS Lambda functions) bằng cách chạy các script sau:  
*cd $CURRENT\_DIR/orchestration/; ./create-orchestration.sh*

**Cấu hình Amazon Cognito user pool**

Trong phần này, chúng tôi tạo một người dùng trong Amazon Cognito user pool của chúng tôi. Người dùng này sẽ đăng nhập vào ứng dụng web của chúng tôi.

Chạy script create-cognito-testuser.sh để tạo người dùng (đảm bảo cung cấp địa chỉ email của bạn):

cd $CURRENT\_DIR/cognito; ./create-cognito-testuser.sh #your-email-address#

Sau khi bạn tạo người dùng, bạn sẽ nhận được một email với mật khẩu tạm thời theo định dạng này: "Your username is #your-email-address# and temporary password is #temporary-password#."

Ghi chú những chi tiết đăng nhập này (địa chỉ email và mật khẩu tạm thời) để sử dụng sau khi thử nghiệm ứng dụng web.

**Tạo ứng dụng web**

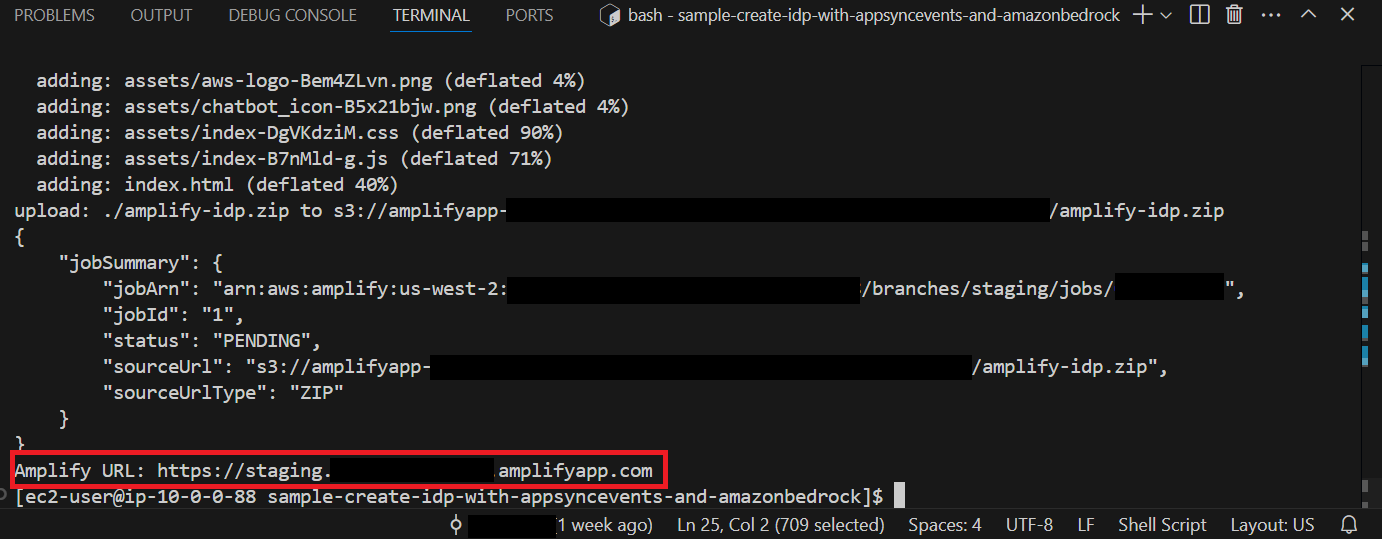
Trong phần này, chúng tôi xây dựng một ứng dụng web sử dụng AWS Amplify và xuất bản nó để làm cho nó có thể truy cập thông qua một URL endpoint.

Hoàn thành các bước sau để tạo ứng dụng web:

1. Chạy script create-webapp.sh để tạo ứng dụng web với AWS Amplify:  
*cd $CURRENT\_DIR/amplify/; ./create-webapp.sh*

2. Chạy script deploy.sh để triển khai ứng dụng web  
*cd $CURRENT\_DIR/amplify/amplify-idp; ./deploy.sh*

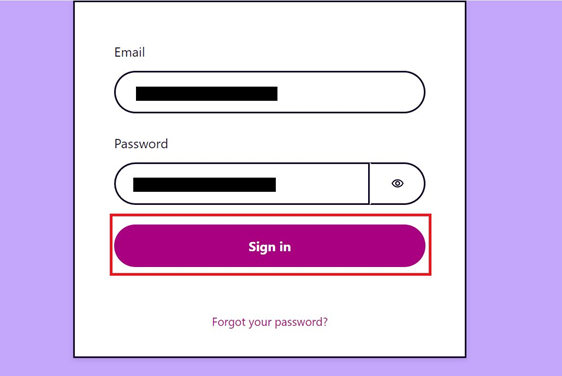
Ứng dụng web hiện có sẵn để thử nghiệm và một URL sẽ được hiển thị, như được hiển thị trong ảnh chụp màn hình sau. Ghi chú URL để sử dụng trong phần sau.



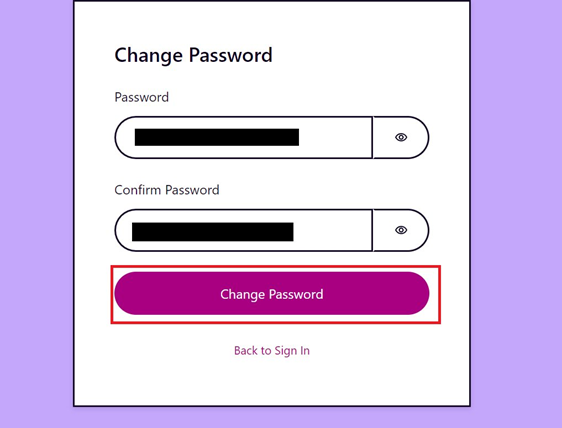
**Thử nghiệm ứng dụng web**

Trong phần này, chúng tôi thử nghiệm ứng dụng web và tải lên một tệp để được xử lý:

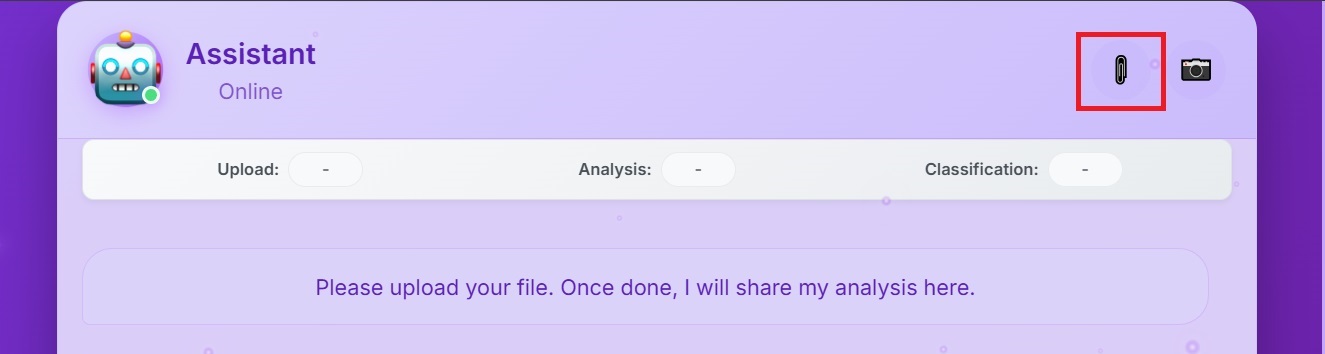
1. Mở URL của ứng dụng AWS Amplify trong trình duyệt web của bạn.  
2. Nhập thông tin đăng nhập của bạn (email và mật khẩu tạm thời bạn đã nhận trước đó khi cấu hình user pool trong Amazon Cognito) và chọn Sign in.



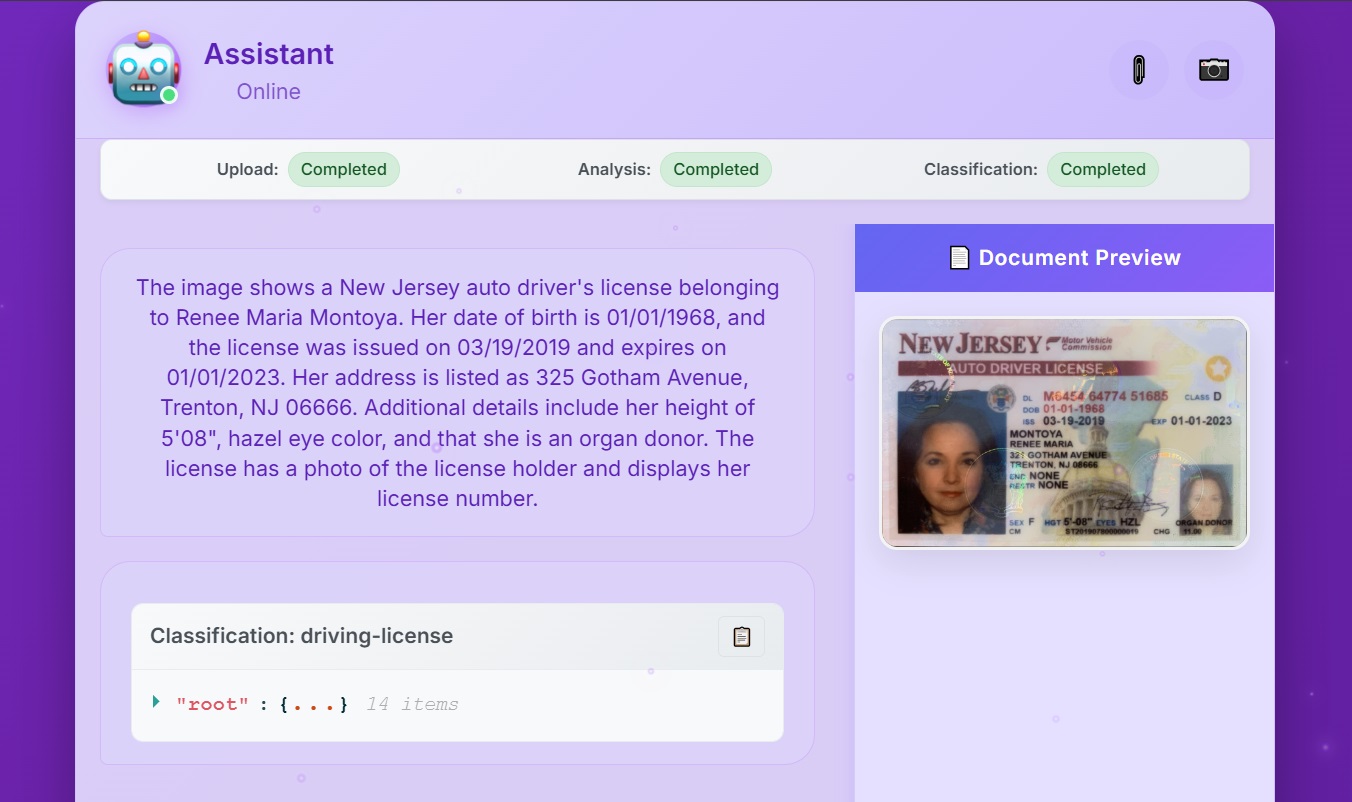
3. Khi được nhắc, nhập mật khẩu mới và chọn Change Password.



4. Bây giờ bạn sẽ có thể thấy một giao diện web.  
5. Tải xuống bằng lái xe mẫu tại vị trí này và tải lên nó qua ứng dụng web sử dụng camera của bạn hoặc một tệp trong thiết bị cục bộ của bạn, như được minh họa



Một khi tệp được tải lên, bạn sẽ bắt đầu nhận phản hồi trong ứng dụng web. Khi tất cả các hoạt động hoàn thành, bạn sẽ thấy một kết quả tương đương với những gì được hiển thị trong ảnh chụp màn hình sau:



Lưu ý: Nếu bạn đang lên kế hoạch sử dụng các hình ảnh bằng lái xe mẫu khác với các định dạng khác, bạn có thể phải cập nhật blueprint Bedrock Data Automation hiện có mà chúng tôi đã tạo trước đó hoặc định nghĩa một blueprint mới trong dự án Bedrock Data Automation mà chúng tôi đã tạo trước đó để những hình ảnh mới này hoạt động. Để biết thêm thông tin, vui lòng xem lại tài liệu Bedrock Data Automation.

**Dọn dẹp**

Để đảm bảo rằng không có chi phí bổ sung nào phát sinh, hãy xóa các tài nguyên được cung cấp trong tài khoản của bạn. Đảm bảo bạn đang ở đúng tài khoản AWS trước khi xóa các tài nguyên sau.

Lưu ý quan trọng: Bạn nên thận trọng khi thực hiện các bước trước. Đảm bảo bạn đang xóa các tài nguyên trong đúng tài khoản AWS.

Bạn có thể điều hướng đến bảng điều khiển AWS CloudFormation để xóa các CloudFormation stacks liên quan đến các tài nguyên được cung cấp hoặc sử dụng script helper dọn dẹp cleanup.sh có sẵn tại gốc của thư mục sample-create-idp-with-appsyncevents-and-amazonbedrock:

*./cleanup.sh #region#*

**Kết luận**

Trong bài viết này, chúng tôi đã đi qua một giải pháp để tạo một đường ống xử lý tài liệu, với một ứng dụng web sử dụng các dịch vụ serverless. Qua ứng dụng web, chúng tôi đã có thể tải lên một tệp và nhận phản hồi theo thời gian thực cho các loại hoạt động khác nhau (tóm tắt, trích xuất các trường cụ thể và phân loại). Đầu tiên, chúng tôi đã tạo một dự án Amazon Bedrock Data Automation (với một blueprint bằng lái xe). Sau đó chúng tôi đã tạo một web socket cùng với một giải pháp điều phối sử dụng một state machine (AWS Step Functions và AWS Lambda functions). Chúng tôi cũng đã cấu hình một user pool để cấp cho người dùng quyền truy cập vào ứng dụng web. Cuối cùng, chúng tôi đã tạo frontend của ứng dụng web trong AWS Amplify.

Để đi sâu hơn vào giải pháp này, một workshop tự học có sẵn trong AWS Workshop Studio.

──────────────────────────────────────────────────

## 📖 Glossary - Thuật ngữ

|  |  |  |
| --- | --- | --- |
| English | Tiếng Việt | Định nghĩa |
| Auto Scaling | Tự động mở rộng quy mô | Khả năng tự động tăng/giảm resources dựa trên demand |
| Load Balancer | Bộ cân bằng tải | Phân phối traffic đến multiple servers |
| Microservices | Kiến trúc microservices | Architectural pattern chia application thành small services |

## 🔗 Tài liệu tham khảo

**Tài liệu gốc**

• AWS Blog: https://aws.amazon.com/vi/blogs/compute/orchestrating-document-processing-with-aws-appsync-events-and-amazon-bedrock/

──────────────────────────────────────────────────

## 💬 Ghi chú của người dịch

**Challenges trong quá trình dịch**

• Technical Terms: Giữ nguyên các thuật ngữ AWS để đảm bảo tính chính xác

• Cultural Context: Điều chỉnh ví dụ cho phù hợp với bối cảnh Việt Nam

• Complex Concepts: Giải thích rõ ràng các khái niệm phức tạp

**Insights gained**

• Technical Learning: Hiểu sâu hơn về kiến trúc AWS và best practices

• Language Skills: Cải thiện kỹ năng dịch thuật kỹ thuật

• Industry Knowledge: Nắm bắt xu hướng công nghệ mới

──────────────────────────────────────────────────

## 🤝 Đóng góp và Feedback

Bài dịch này được thực hiện trong khuôn khổ FCJ Internship Program.

**📧 Liên hệ:** ngqbinh456@gmail.com.  
**💬 Feedback:** Mọi góp ý để cải thiện chất lượng dịch thuật xin gửi về email trên  
**🔄 Updates:** Bài dịch sẽ được cập nhật dựa trên feedback từ cộng đồng

──────────────────────────────────────────────────

© 2024 - Bản dịch thuộc về Nguyễn Quang Bình. Vui lòng credit khi sử dụng.