

# Automated Content Sync Solution

## Overview

The goal is to automate the process of fetching, packaging, and syncing content between Adobe Experience Manager (AEM) instances using Query Builder queries and Curl, and to host the sync utility on an EC2 instance for scalability and ease of management.

## Problem Statement

Content synchronization between AEM instances (e.g., from development to staging or production) is often manual and error-prone. Automating this process can enhance efficiency, reduce errors, and ensure consistency across environments.

## Proposed Solution

### Steps

1. **Utilize AEM Query Builder:** Leverage Query Builder to fetch the required content paths based on specific criteria.
2. **Automate Package Creation:** Programmatically create and build AEM packages using the fetched content.
3. **Sync Content Using Curl:** Transfer the created packages between AEM instances using Curl commands.
4. **Host Sync Utility on EC2 Instance:** Deploy the sync utility on an EC2 instance for automated and scalable execution.

## Implementation

1. **Set Up Query Builder:** Define Query Builder parameters in a JSON file to specify the criteria for fetching content and execute the query using Curl.
2. **Create a Package Programmatically:** Write a script to fetch query results and create an AEM package, then build the package.
3. **Sync Content Using Curl:** Download the created package and upload it to another AEM instance using Curl commands.
4. **Host Sync Utility on EC2 Instance:** Create an EC2 instance, convert the Bash script into a Python-based solution for ease of integration, and deploy it on the instance.

# Pros and Cons

## Pros

- **Efficiency:** Automates repetitive tasks, saving time and reducing human errors.
- **Consistency:** Ensures content synchronization is consistent across different AEM instances.
- **Scalability:** Hosting the sync utility on an EC2 instance allows for scalable and on-demand execution.
- **Manageability:** Simplifies management and execution of the sync process, reducing operational overhead.
- **Flexibility:** Customizable and adaptable to specific needs and environments.

## Cons

- **Complexity:** Requires setting up and maintaining EC2 instances and scripting for package creation and sync.
- **Initial Setup:** Involves a learning curve and initial configuration effort.
- **Dependency on External Services:** Relies on EC2 instances and proper internet connectivity for execution.

## Conclusion

By implementing this automated content sync solution using Query Builder, Curl, and EC2 instances, we can achieve a streamlined, efficient, and scalable process for managing content synchronization between AEM instances. This PoC demonstrates the feasibility and benefits of the proposed approach, paving the way for a full-scale implementation in production environments.

*Created by Bhuwan Bhaskar on 26th July 2024*