Implementation Guide

Amazon Marketing Cloud Uploader from AWS

Amazon Marketing Cloud Uploader from AWS: Implementation Guide

Table of Contents

Solution overview **1**
- Features and benefits 2
- Use cases 4
- Concepts and definitions 4

Architecture overview **5**
- Architecture diagram 5
- AWS Well-Architected design considerations 6
  - Operational excellence 7
  - Security 7
  - Reliability 8
  - Performance efficiency 8
  - Cost optimization 9
  - Sustainability 9

Architecture details **10**
- Web application 10
- Amazon API Gateway 10
- Amazon DynamoDB 10
- AWS Glue 10
- AWS Lambda 10
- Amazon S3 11
- Amazon CloudFront 11
- AWS Secrets Manager 11
- AWS services in this solution 11
- How Amazon Marketing Cloud Uploader from AWS works 12

Plan your deployment **14**
- Supported AWS Regions 14
- Cost 14
  - Sample cost table 15
- Security 16
  - IAM roles 16
  - Amazon CloudFront 16
  - AWS Secrets Manager 16
  - AWS CloudTrail 17
  - Multi-factor authentication (MFA) in Amazon Cognito user pools 17

AWS Web Application Firewall (WAF) in Amazon API Gateway 17
Securing log files 17
Use CloudTrail to activate logging, auditing, and alerting 17
Secure logging in API Gateway for AWS service APIs 18
Redact sensitive data from CloudTrail logs 19
Log retention 20
AWS CloudFormation parameters 20
Quotas 21
Quotas for AWS services in this solution 21
AWS CloudFormation quotas 21
Deploy the solution 22
Prerequisites 22
Deployment process overview 22
AWS CloudFormation template 23
Step 1: Set up first-party data S3 bucket 24
Optional: KMS encryption key usage 24
Set permissions for the S3 bucket 25
Step 2: Upload your first-party data to S3 26
Step 3: Launch the stack 26
Step 4: Access the web interface 28
Identify the web interface URL 28
Step 5: (optional) Create user accounts 28
Monitor the solution 31
Activate CloudWatch Application Insights 31
Confirm cost tags associated with the solution 33
Activate cost allocation tags associated with the solution 34
AWS Cost Explorer 35
Update the solution 36
Troubleshooting 38
Problem: I have not received my temporary password to the web UI 38
Resolution 38
Problem: AWS Glue job has status FAILED 38
Resolution 38
Problem: CloudFormation stack deployment fails 39
Resolution 39
Problem: Why did so few identities resolve for my dataset? 39

Resolution 39
Contact Support 39
Create case 40
How can we help? 40
Additional information 40
Help us resolve your case faster 40
Solve now or contact us 40
Uninstall the solution 41
Using the AWS Management Console 41
Using AWS Command Line Interface 41
Use the solution 42
Allow a Return URL 42
Specify AMC instances 42
Select files 43
Select AMC destinations 44
Define the dataset 45
Define the schema 47
Confirm details 49
Monitor job and verify dataset successfully uploaded 50
Developer guide 51
Source code 51
Customization guide 51
Retrieving Client ID and Client Secret 51
Locating AMC instance information 51
OAuth Flow in Amazon Ads API 52
AMC data upload file format requirements 52
CSV file requirements 52
JSON file requirements 53
AMC data types, timestamp, and date formats 53
AMC FACT compared with DIMENSION datasets 54
Fact datasets 54
Dimension datasets 55
Reference 56
Anonymized data collection 56
Contributors 57
Revisions 58

Notices 59

Use this solution to upload first-party signals into Amazon Marketing Cloud (AMC) for evaluating and planning advertising campaigns

Advertisers and their partners have asked for easier ways to generate insights from their collective signals to plan, activate, and measure advertising campaigns. These customers look to AWS for guided workflows and automation tools to simplify multi-party collaboration *clean rooms* and accelerate consumer insights for their advertising use cases.

Amazon Marketing Cloud (AMC) is a secure, privacy-safe, and dedicated cloud-based environment in which advertisers can easily perform analytics across multiple, pseudonymized signals to generate aggregated reports. Inputs can include an advertiser’s own signals, as well as their Amazon Ads campaign events, such as impressions, clicks, and conversions. AMC reports can help with campaign measurement, audience refinement, supply optimization, and more, allowing advertisers to make more informed decisions about their cross-channel marketing investments.

Using first-party signals is crucial to companies' advertising efforts, including in Amazon Ads, yet many brands face challenges in going from signal source to formatting and uploading pseudonymous signals. Brands have limited technical resources and need ready-to-use solutions to remove the heavy lifting of normalizing, hashing, and preparing data for uploading to AMC.

The Amazon Marketing Cloud Uploader from AWS solution helps Amazon Ads customers seamlessly upload customer signals into their Amazon Marketing Cloud (AMC) instance without dedicating IT resources to build and support the upload workflows. This allows Amazon Ads customers to continue optimizing their Amazon Ads campaigns within AMC while maintaining complete control of their data, as output data is encrypted, transferred, and normalized before it is uploaded into the AMC instance.

This solution is available today to Amazon Ads customers with an existing AMC instance and access to the AWS Account associated with AMC. Customers can use an IAM Role or credential generated in their AWS account to access the AMC API. Customers can deploy this solution directly from the AWS Solutions library using the AWS CloudFormation template to run the normalization, hashing, file transfer, and API calls required by the AMC API. After data is uploaded users can query the AMC API for multi-party collaboration insights across first-party and Amazon Ads' pseudonymous signals.

This guide is intended for solutions architects, DevOps engineers, data scientists, and cloud professionals who want to implement Amazon Marketing Cloud Uploader from AWS in their environment.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution. The estimated cost for uploading 1 terabyte per month with this solution in the US East (N. Virginia) Region is **USD $543.24 per month**.	Cost
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions support this solution.	Supported AWS Regions
View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.	GitHub repository

Features and benefits

The Amazon Marketing Cloud Uploader from AWS solution provides the following features:

Marketers and their partners can:

1. Ensure first-party signals are stored, encrypted, and controlled within your AWS account and have end-to-end visibility on each step of the activation workflow.

2. Improve campaign planning by developing insights based on multi-party customer intersection and attribute enrichment.
3. Provide campaign measurement and attribution with multi-party data that helps connect the dots in a consumer's path to purchase from Amazon Ads properties to customer's first-party and third-party signals.

Amazon Marketing Cloud Uploader from AWS transforms data to streamline ingestion into AMC

This solution transforms the data for ingestion into AMC to meet AMC's unique requirements on the data before ingestion.

User interface for transformation of data

AMC requires a specific format prior to ingestion. Amazon Marketing Cloud Uploader from AWS provides a user interface (UI) via a webpage to define the transformations.

API to automate the transformation of data and loading into AMC

An API is available for customers looking to automate the process of transforming data, and ingestion of the transformed data into AMC.

Amazon Marketing Cloud Uploader from AWS is launched in the customer AWS account so customers have full control of the data they want to share

Users retain control over the data they're interested in sharing with AMC.

Amazon Marketing Cloud Uploader from AWS also serves as a reference application from which developers can jump-start their own custom ETL pipelines for AMC

Developers can modify the data normalizations in the provided AWS Glue job to maximize identity resolution in AMC for their organization's datasets.

Integration with AWS Service Catalog AppRegistry and Application Manager, a capability of Systems Manager

This solution includes an AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both AppRegistry and Application Manager. With this integration, you can centrally manage the solution's resources and enable application search, reporting, and management actions.

Use cases

Amazon Marketing Cloud Uploader from AWS users can analyze their data alongside AMC data to gain insights into the customer journey

Amazon Marketing Cloud Uploader from AWS makes it easier to upload data to AMC, so customers can gain insight from the combined customer and AMC data.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

Personally identifiable information (PII)

PII is a textual reference to personal data that could be used to identify an individual. PII examples include addresses, bank account numbers, and phone numbers.

Amazon Marketing Cloud (AMC)

Amazon Marketing Cloud (AMC) is a secure, privacy-safe, and cloud-based clean room solution, in which advertisers can easily perform analytics across pseudonymized signals, including Amazon Ads signals as well as their own inputs.

> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary.

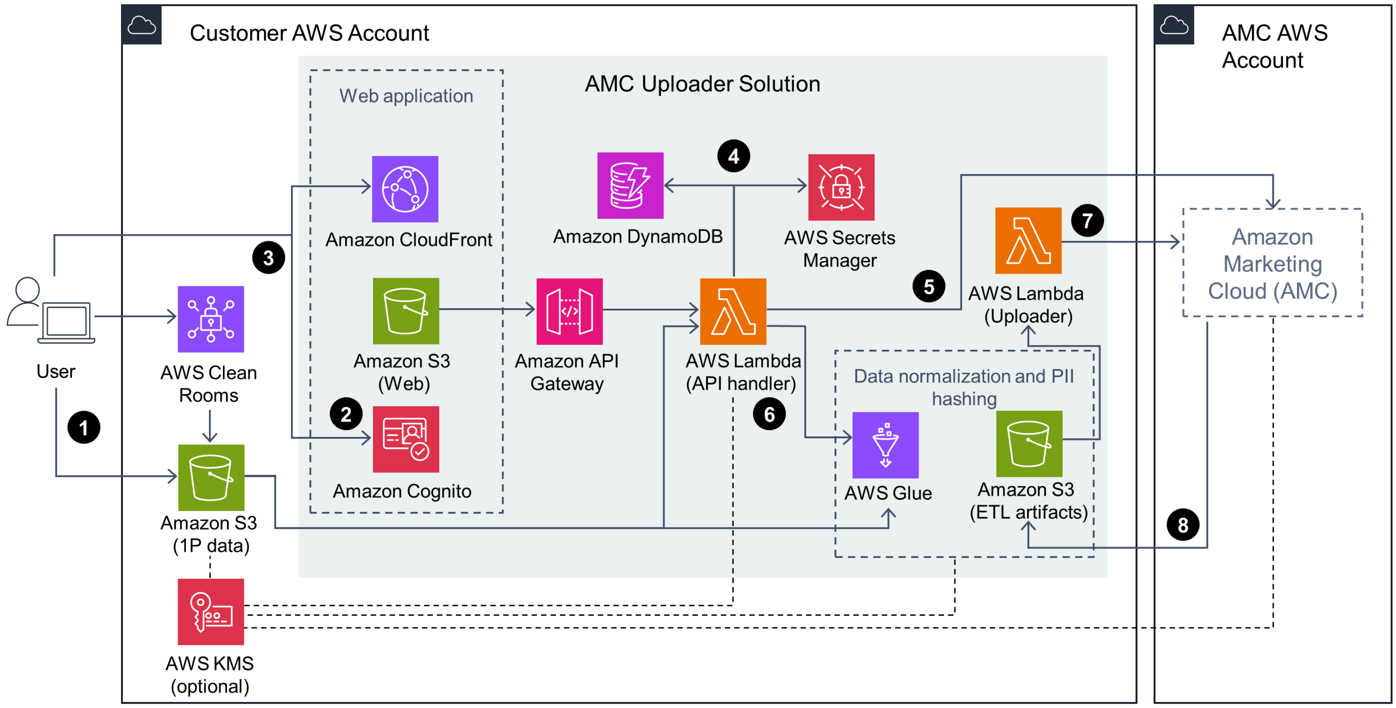
Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.

Amazon Marketing Cloud Uploader from AWS architecture diagram



The high-level process flow for the solution components deployed with the AWS CloudFormation template is as follows:

1. User uploads first-party data to designated Amazon Simple Storage Service (Amazon S3) bucket or exports data from AWS Clean Rooms to the S3 bucket. Optionally, the user can designate data from the S3 bucket to an AWS Key Management Service (AWS KMS) key to decrypt and encrypt source data and its derivatives throughout the extract, transform, load (ETL) pipeline.

2. User logs in with Amazon Cognito to the provided web application and obtains the authorization tokens needed to load frontend assets from Amazon S3 and backend resources from Amazon API Gateway.
3. Users interact with the provided web application through an Amazon CloudFront distribution and an API Gateway endpoint. The CloudFront resource serves static website assets from Amazon S3. The API Gateway resource provides a REST API interface to the API handler AWS Lambda resource. This resource includes a variety of functions for creating, reading, updating, and deleting datasets. When an AWS KMS key is used to encrypt first-party data, this resource will also use the designated AWS KMS key to decrypt and read that data.
4. The Amazon DynamoDB resource stores system configurations, such as user-specified connection details for Amazon Marketing Cloud instances. These configurations are inputs in the frontend web form and saved by the API handler Lambda resource. User-specified OAuth credentials are saved to AWS Secrets Manager as well as the programmatically-derived OAuth refresh token.
5. The API handler Lambda resource interacts with one or more Amazon Marketing Cloud instances in order to create, read, update, and delete datasets.
6. When users submit requests to upload data to new or existing datasets, the API handler Lambda resource starts an AWS Glue ETL job to normalize, hash, and reformat user-specified files according to the data upload rules of Amazon Marketing Cloud. The AWS Glue job will use the optionally designated AWS KMS key to decrypt the first-party data and encrypt transformed data objects when they are written to Amazon S3.
7. The AWS Glue job outputs results to an ETL artifacts Amazon S3 bucket. This event initiates a request from the Uploader Lambda resource to each user-specified Amazon Marketing Cloud instance to initiate uploads of those results.
8. Each user-specified Amazon Marketing Cloud instance asynchronously uploads transformed data objects from the ETL artifacts Amazon S3 bucket and uses the optionally designated AWS KMS key to decrypt those objects when needed.

AWS Well-Architected design considerations

This solution was designed with best practices from the AWS Well-Architected Framework, which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

- **Operations as code** - A REST API provides the ability to run ETL workflows by invoking them on-demand, on a schedule, or in response to events.
- **Refine operations procedures** - A modular data transformation resource provides the opportunity to refine data normalization logic to improve match rates in AMC.
- **Application telemetry** - Metrics and logs stored in AWS CloudWatch and AWS X-Ray provide insight into user activity, end-to-end transactions, and the health of back-end resources.
- **Design for operations** - Versioning in solution source allows tracking of changes and releases. Versioning in pre-built AWS CloudFormation templates and the built-in REST API allows end users to revert to known good states, previous versions, and limit the risk of assets being lost.
- **Continuous improvement** - A public GitHub repository provides a forum where users and developers can collaborate to address software defects and feature requests.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- Amazon Cognito provides authentication, authorization, and user management for the web application.
- Access to each resource in the infrastructure is controlled by AWS Identity and Access Management (IAM).
- All IAM policies have been scoped down to the minimum permissions required for the service to function properly.
- AWS Secrets Manager stores user-specified OAuth credentials.
- HTTP clients obtain the permissions needed to run their requests by providing access keys and tokens through the Signature Version 4 (SigV4) signing process.
- SigV4 signing is integrated into the web application to map Amazon Cognito tokens to IAM policies.

- All data storage including Amazon S3 buckets have encryption at rest. All data in motion is encrypted using Transport Layer Security (TLS).
- Amazon CloudFront improves website security with traffic encryption and access controls, and can use AWS Shield Standard to defend against distributed denial-of-service (DDoS) attacks at no additional charge.
- This solution went through Amazon application security reviews before release.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

- The solution exclusively uses AWS serverless services (for example, AWS Lambda, Amazon API Gateway, and Amazon S3) to ensure high availability and recovery from service failure.
- Data processing uses AWS Lambda functions. Data is stored in Amazon S3, so it persists in multiple Availability Zones (AZs) by default; Amazon S3 offers 99.999999999% (11 9s) durability.
- Amazon CloudFront is used to render static content from the user's AWS account to a publicly available website. Amazon CloudFront reduces latency by delivering data through 410+ globally dispersed Points of Presence (PoPs) with automated network mapping and intelligent routing.
- The use of AWS Lambda functions as a serverless architecture with no dedicated VMs, which increases reliability through distributed computing.
- The solution automatically throttles Lambda functions through concurrency settings when connected to downstream API endpoints to meet service quota limits.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- The solution uses serverless architecture throughout.
- The solution is periodically reviewed by solution architects and subject matter experts for areas to experiment and improve.
- AWS Glue is an efficient ETL engine for running scripts and transforming the data.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- The solution uses serverless architecture therefore, customers only get charged for what they use.
- The compute layer defaults to AWS Lambda, so it provides pay-per-use.
- AWS Glue jobs for ETL are batched in the largest size allowed to minimize transaction costs.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

- The solution utilizes managed and serverless services, to minimize the environmental impact of the backend services. The solution’s serverless design using AWS Lambda, Amazon S3, and the use of managed services such as Amazon Cognito, are aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

Web application

After deploying via AWS CloudFormation template, the web application allows the user to select a file to be used as the source data and provides a way to define the transformations of the selected data prior to ingestion by Amazon Marketing Cloud.

Amazon API Gateway

Dynamic functionality is implemented in the static web application by using JavaScript to call a REST API built with AWS Lambda and Amazon API Gateway.

Amazon DynamoDB

Within the provided web application, users must specify endpoint attributes for one or more target AMC instances. These properties are stored in an Amazon DynamoDB table and used to establish connections for creating and populating datasets.

AWS Glue

AWS Marketing Cloud requires the data be in a specific format prior to ingestion. AWS Glue is the service that works with the web client to normalize and transform the data as defined in the user interface. Included in that transformation is the ability to define the time-series partition of the data (hour, day, month). After the AWS Glue job is complete, a notification is sent using Amazon S3 event notifications.

AWS Lambda

There are two Amazon Lambda resources in this solution. One Lambda function acts as the API handler to process client requests. Another Lambda function notifies AMC to begin the ingestion process when datasets are ready for upload.

Amazon S3

Two Amazon S3 buckets are used, one for hosting static web application resources, and one for storing ETL artifacts. AMC is notified to begin uploading data transformation results as soon as they are written to the ETL artifacts bucket. The ETL artifacts bucket has a lifecycle policy to automatically remove data transformation results after three days. This setting can be modified in the AWS console under the S3 settings. For details, refer to Setting lifecycle configuration on a bucket in the *Amazon Simple Storage Service User Guide*.

Amazon CloudFront

This solution deploys a web console hosted in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to the solution's website bucket contents. For more information, refer to Restricting access to an Amazon S3 origin in the *Amazon CloudFront Developer Guide*.

AWS Secrets Manager

Within the provided web application, users enter client ID and client secret credentials along with other connection details for each AMC instance. These OAuth credentials are stored in AWS Secrets Manager and used to perform authorization with AMC.

AWS services in this solution

AWS service	Description
AWS Glue	**Core**. AWS Glue transforms and normalizes the data in preparation for ingestion into AMC.
AWS Lambda	**Core**. One Lambda function creates the dataset within the AMC instance via API Gateway. The other Lambda function calls the AMC API to begin the ingestion process.
Amazon S3	**Core**. S3 hosts the web client, first party data, and the ETL artifact data.

AWS service	Description
Amazon API Gateway	**Supporting**. Provides a way to run the application via API, or via the web client.
Amazon CloudFront	**Supporting**. CloudFront improves security with traffic encryption and works with Amazon S3 and Amazon Cognito on access control.
Amazon Cognito	**Supporting**. Provides authorization of users to web client.
Amazon DynamoDB	**Supporting**. Uploads to one or more AMC instances.
AWS Secrets Manager	**Supporting**. Saves OAuth credentials.

How Amazon Marketing Cloud Uploader from AWS works

The AMC data upload functionality includes very specific requirements for the following data preparation criteria:

1. Normalization
2. PII hashing
3. File partitioning by size

The primary objective of this solution is to help AMC users prepare data files according to those criteria and to subsequently load them into new AMC datasets.

Amazon Marketing Cloud Uploader from AWS provides a web application that guides users to specify the information required for dataset definitions and data preparation. Once users submit their information through the web application, an AWS Glue job transforms their data files according to AMC's data preparation criteria and saves the resulting files to an ETL artifact bucket.

Although this solution automates the process of data preparation, the original files must meet a set of file formats. Refer to the AMC data upload file format requirements section.

The AWS Glue job can fail for the following two reasons:

1. If the original data files are not formatted according to the specification in the Prepare data documentation, then the AWS Glue job will explicitly fail.
2. If original data files include addresses, phone numbers, or other values which require normalization that has not been implemented in the AWS Glue job then AMC will not be able to use those fields to resolve identities.

You can identify errors which lead to explicit AWS Glue job failures from the job logs provided in the AWS Glue console.

If you observe poor identity resolution rates in AMC from datasets that you uploaded using this solution, then you should open the AWS Glue job output files to validate whether your data has been normalized according to the rules documented in the Prepare data documentation.

Note

The location for the output files will be shown in the AWS Glue job run log.

Plan your deployment

This section describes the cost, security, log retention, and quota considerations for planning your deployment.

Supported AWS Regions

This solution uses services, which are not currently available in all AWS Regions. For the most current availability of AWS services by Region, see the AWS Regional Services List.

This solution is supported in the following AWS Regions:

Region Name	
US East (N. Virginia)	Asia Pacific (Sydney)
US East (Ohio)	Canada (Central)
US West (N. California)	Europe (Ireland)
US West (Oregon)	Europe (London)
Asia Pacific (Mumbai)	Europe (Paris)
Asia Pacific (Tokyo)	Europe (Stockholm)
Asia Pacific (Seoul)	Europe (Frankfurt)
Asia Pacific (Osaka)	South America (São Paulo)
Asia Pacific (Singapore)	

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the estimated cost for uploading 1 terabyte per month with this solution in the US East (N. Virginia) region is USD $543.24 per month.

Refer to the pricing webpage for each AWS service used in this solution.

We recommend utilizing AWS Budget Alerts and monitoring costs through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

No licenses are required to deploy this solution. There is no cost to use this solution, but you will be billed for any AWS services or resources that this solution deploys.

Sample cost table

Example scenario: A customer wishes to send a transform and ingest sales data with 1,000 rows to AMC, once a day.

The following table provides a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month.

AWS service	Dimensions	Cost [USD]/month
AWS Key Management Service	Two KMS keys per stack and encryption operations from uploading 37 GB per day .	$2.30
Amazon S3	Temporary storage required to upload 37GB per day.	$0.50
Amazon API Gateway	API requests required to handle 100 uploads per day.	$0.01
AWS Lambda	Compute costs required to handle 100 uploads per day.	$0.01
Amazon DynamoDB	1 user performing 10 uploads per day interactively on the AMC Uploader web interface.	$0.02
AWS Glue	41 DPU `hoursforA pacheSpark` Job to upload 37GB per day.	$540.00

AWS service	Dimensions	Cost [USD]/month
AWS Secrets Manager	1 secret a month.	$0.40
	Total:	**$543.24**

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit AWS Cloud Security.

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda functions access to create regional resources.

Amazon CloudFront

This solution deploys an Amazon CloudFront distribution and uses the default CloudFront domain name and SSL certificate. The default CloudFront SSL certificate only supports TLSv1. To use a later TLS version (TLS1.2 and above), use your own domain name and custom SSL certificate. For more information, refer to Using alternate domain names and HTTPS in the *Amazon CloudFront Developer Guide*.

This solution deploys a web client hosted in an Amazon Simple Storage Service (Amazon S3) bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is an Amazon CloudFront user that provides public access to the solution's website bucket contents. For more information, refer to Restricting access to an Amazon S3 origin in the *Amazon CloudFront Developer Guide*.

AWS Secrets Manager

This solution uses AWS Secrets Manager to securely store user-specified OAuth credentials.

AWS CloudTrail

If your company must comply with SOC (Systems and Organization Controls), PCI DSS (Payment Card Industry Data Security Standard), HIPAA (Healthcare Information Portability and Accountability Act), or any other regulation, it is your responsibility to ensure compliance by activating AWS CloudTrail for secure logging as required by your organization's security policy.

Multi-factor authentication (MFA) in Amazon Cognito user pools

This solution creates only one user in its Amazon Cognito user pool. MFA is not activated by default; however, we recommend using MFA for users in Amazon Cognito for a stronger security posture in production workloads. For more information about setting up MFA in Amazon Cognito, refer to Adding MFA to a user pool and Adding advanced security to a user pool in the *Amazon Cognito Developer Guide*.

AWS Web Application Firewall (WAF) in Amazon API Gateway

We recommend activating AWS WAF for the Amazon API Gateway for this solution when the application is open to public in production environment. For guidance about setting up WAF, refer to Using AWS WAF to protect your APIs in the *Amazon API Gateway Developer Guide*. We also recommend reviewing the AWS Best Practices for DDoS Resiliency whitepaper for information about protecting your AWS applications from Distributed Denial of Service (DDoS) attacks.

Securing log files

Log files are a potential security vulnerability that should be mitigated as thoroughly as possible. The following are AWS best practices for securing log files:

Use CloudTrail to activate logging, auditing, and alerting

CloudTrail must be activated in all AWS accounts for all AWS products. This helps with security auditing in case of a security incident. Development stage accounts must also have CloudTrail activated as development environments are frequently attacked on the assumption that their security controls are weaker than those of production environment.

Refer to the AWS CloudTrail User Guide for instructions on activating it.

Often when a compromise happens, actors try to enumerate permissions on a compromised IAM user or role, which will generate authorization failures. We recommend Monitoring CloudTrail Log Files with Amazon CloudWatch Logs.

If your case requires strong integrity guarantees, consider activating CloudTrail Log File Validation feature.

All AWS accounts must have CloudTrail activated and alerting setup.

- Verify that CloudTrail is activated in all Regions
- Verify that S3 bucket where CloudTrail logs are stored is locked down
  - Use scoped down bucket policy that gives service operators permissions to read but not write to the bucket (log records must be written only once and stay immutable)
- Verify that alerts function properly
  - Perform an action that will generate an **UnauthorizedOperation** or **AccessDenied** error in CloudTrail logs
  - Confirm that the alert has been invoked and received
- Verify that CloudTrail Log File Integrity is activated
- Verify that at least one trail of CloudTrail in each account captures events from global services, such as IAM and AWS Security Token Service (STS). You can activate global service events logging for a trail from AWS CLI by running the following command:

```
update-trail --name <trail_name> --include-global-service-events
```

Secure logging in API Gateway for AWS service APIs

When execution logs are activated for an API Gateway stage, API Gateway redacts authorization header values, API key values, and similar sensitive request parameters. If you turn on **Full request and response logs**, API Gateway logs full request and response execution logs. Do not expose customer data or sensitive information in logs. We recommend that you do not use **Full request and response logs** for production APIs.

You turn on **Full request and response logs** by setting `dataTraceEnabled` to `True`. If you want to activate the full request and response logs, consult a security engineer before doing so. Otherwise, activate access logging and annotate only non-sensitive parameters. To set up access logs, see Setting up CloudWatch logging for a REST API in API Gateway.

We recommend the following best practices for secure logging in API Gateway for AWS service APIs:

- Turn off execution logging in API Gateway.
- Turn on access logging in API Gateway.
- If you are using AWS CloudFormation, set the `dataTraceEnabled` parameter to `True`.
- Inspect your CloudWatch Logs for the API Gateway endpoint and verify that full request response objects are not logged.
- If you must turn on execution logging, use `INFO` level logs to maintain auditing capability by capturing the required information in the logs.

This solution deploys an Amazon API Gateway REST API and uses the default API endpoint and SSL certificate. The default API endpoint only supports TLSv1. To use a later version of TLS, use your own domain name and custom SSL certificate. For more information, see Choosing a security policy for your custom domain in API Gateway in the *Amazon API Gateway Developer Guide*.

Redact sensitive data from CloudTrail logs

Sensitive data includes PII, passwords, credentials, among others. For more information about what is considered sensitive, refer to your organization's security policy.

Make sure sensitive data fields are redacted in the payload:

1. **Customized redaction** - This is done through cloning the request/response object and stripping the fields with sensitive information within your code.
2. **Automated redaction** - CloudTrail automatically redacts the fields with the **sensitive** trait, hence this trait should be used for sensitive parameters.
3. **Keyword redaction** - An additional useful control is the keyword redaction feature which automatically redacts fields that have specific keywords in their names which could indicate they are sensitive (for example, password). Note that you shouldn't solely depend on this feature, but you must only use it as an additional layer besides the two options mentioned previously.

Review the request parameters and the response elements for the events that you are logging to CloudTrail with your AppSec security engineer, and make sure all sensitive fields are redacted.

Request parameters and the response elements for the events you are logging to CloudTrail don't contain any sensitive data.

Log retention

The history of a malicious user's activities might be lost if logs are not retained for a long enough period of time.

Store your security relevant logs in accordance with your organization's security policy. Your log retention period might vary depending on your specific needs. Discuss with AppSec cases where you're considering reducing your log retention to make sure that you will have the data you need to investigate and reconstruct events long after they occur.

AWS Security recommends storing your logs for 10 years unless there is a good reason not to, and you have validated that reason with your organization's security policy.

Store your logs remotely (away from the generator of those logs) in a secure environment. For example, logs can be stored in an S3 bucket and later migrated to Amazon S3 Glacier.

All security relevant logging is kept for ten years unless there is a really good reason not to, and you have validated that reason with your organization's security policy.

AWS CloudFormation parameters

- **AdminEmail** - Email address of the solution administrator.
- **DataBucketName** - Name of the S3 bucket from which source data will be uploaded.
- **CustomerManagedKey** - (Optional) Customer Managed Key to be used for decrypting source data, encrypting ETL results, and encrypting the corresponding datasets in AMC. The **CustomerManagedKey** must be located in the same Region as the CloudFormation stack.
- AMC provides the ability to encrypt customer datasets with encryption keys created in AWS Key Management Service (KMS). This step is optional. If an encryption key is not provided, AMC will perform default encryption on behalf of the customer. The benefit to using a customer generated encryption key is the ability to revoke AMC's access to uploaded data at any point. In addition, customers can monitor encryption key access via AWS CloudTrail event logs.
- To activate this feature, specify a key in the **CustomerManagedKey** CloudFormation parameter and modify the key's policy to grant usage permissions to the AMC instance. For more information, refer to KMS Encryption Key Usage documentation.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the AWS services implemented in this solution. For more information, refer to AWS service quotas.

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas PDF page.

AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, refer to AWS CloudFormation quotas in the *AWS CloudFormation User's Guide*.

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Prerequisites

1. An active Amazon Marketing Cloud (AMC) instance. You can create and manage your instance either through the AMC console or through the AMC APIs.
2. Keep note of these parameters from the Amazon Marketing Cloud that will be used throughout this solution. See section Locating AMC instance information to retrieve these values.
    - AMC instance ID
    - Amazon Ads Advertiser ID
    - Data upload AWS account ID
3. Keep note of OAuth credentials from Login with Amazon (LwA) application that will be used by this solution for authorizing to Amazon Ads API. To set up LwA and request access to Amazon Ads API, see Amazon Ads Developer guides and the sections Retrieving Client ID and Client Secret.
    - Client ID
    - Client Secret
4. AWS account with administrator access to create an S3 bucket, IAM roles, and deploy CloudFormation templates.
5. First-party data formatted in CSV or JSON format per the requirements described in the AMC data upload file format requirements section of this guide.

Deployment process overview

Before you launch the solution, review the cost, architecture, security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 10 minutes

Step 1: Set up first-party data S3 bucket

- Create an Amazon S3 bucket

Step 2: Upload your first-party data to S3

- Upload your files

Step 3: Launch the stack

Step 4: Access the web interface

- Open the URL shown in the **UserInterface** output of the base stack. You can also get this URL with the following AWS CLI command:

```
aws cloudformation --region $REGION describe-stacks --stack-name $STACK_NAME --query
 "Stacks[0].Outputs[?OutputKey=='UserInterface'].OutputValue" --output text
```

- Sign in to the web application.

> ⚠ **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the Privacy Notice.
> To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated template and deploy the solution. For more information, refer to the Anonymized data collection section of this guide.

Step 5: (optional) Create user accounts

AWS CloudFormation template

You can download the CloudFormation template for this solution before deploying it.

View template

amazon-marketing-cloud-uploader-from-aws.template - Use this template to launch the solution and all associated components. The default configuration deploys the core and supporting services found in the AWS services in this solution section, but you can customize the template to meet your specific needs.

This CloudFormation template deploys Amazon Marketing Cloud Uploader from AWS in the AWS Cloud. You must meet the prerequisites before launching the stack.

Step 1: Set up first-party data S3 bucket

Set up an S3 bucket and configure it for storing data files to be processed by the Amazon Marketing Cloud Uploader from AWS. The following procedure explains how to create this S3 bucket with the necessary permissions.

1. Log into the AWS Management Console. Ensure you are in the correct Region (refer to the requirements previous section).
2. Navigate to S3 console and select **Create bucket**.
3. Under General configuration, give the bucket a name, e.g., "first-party-data", and select the same AWS region where you will deploy Amazon Marketing Cloud Uploader from AWS. The remaining settings can be left to their default options.
4. Select Create bucket.
5. Save this bucket name for use in Step 3: Launch the stack.

Optional: KMS encryption key usage

Amazon Marketing Cloud Uploader from AWS supports customer encrypted data using AWS Key Management Service (KMS). This is optional. If you do not choose to provide a KMS encryption key, the data will be encrypted at rest by default.

The benefit to using a customer generated encryption key is the ability to revoke AMC's access to previously uploaded data at any point. In addition, customers can monitor encryption key access via AWS CloudTrail event logs.

To use Amazon Marketing Cloud Uploader from AWS with a customer-specified KMS key, follow the procedure described in the KMS Encryption Key Usagesection of the *AMC Developer* guide.

Set permissions for the S3 bucket

Define permissions in the bucket policy that enable the Amazon Marketing Cloud Uploader from AWS to list and read data files from the S3 bucket.

1. Open the S3 bucket, and navigate to the Permissions tab.
2. In the Permissions tab, within Bucket Policy, select **Edit**.
3. Add the following permissions object:

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
      "Effect": "Allow",
      "Principal": {
         "AWS": "arn:aws:iam::{AWS account ID}:root"
       },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket"
       ],
      "Resource": [
        "arn:aws:s3:::{bucket_name}/*",
        "arn:aws:s3:::{bucket_name}"
         ]
       }
    ]
 }
```

- Replace `{bucket_name}` with the name of the newly created bucket.
- Replace `{AWS account ID}` with the AWS account ID of the account where Amazon Marketing Cloud Uploader from AWS will be installed.

Be sure to follow security best practices for Amazon S3 when setting up this bucket.

Step 2: Upload your first-party data to S3

Before uploading your data, make sure your dataset complies with the requirements described in the AMC data upload file format requirements section in this guide. You can also refer to this information in the Prepare Data document or within the *AMC Data Upload Documentation.pdf* document, which can be downloaded from the **Documentation** link shown on your AMC instance administration page.

1. Log into the AWS Management Console. Ensure you are in the correct Region (refer to the requirements in the previous section).
2. Navigate to **S3** and select the bucket you created in Step 1.
3. Choose **Upload.**
4. Depending on how your files or stored, you can either choose **Add Files** to select files on your local computer, or drag and drop files into the highlighted drag and drop area.
5. Once completed, choose **Upload**. Your files have now been uploaded and can be used with the Amazon Marketing Cloud Uploader from AWS web application.

Step 3: Launch the stack

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 10 minutes

1. Sign into AWS Management Console and select the button to launch `advertiser-audience-uploads-to-amazon-marketing-cloud.template` CloudFormation template.

   Launch solution

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. The stack name length must be 32 characters or less.

5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
AdminEmail	`<Requires input>`	Email address for the administrator. This user receives an email with a temporary password to the web application once the AWS CloudFormation template has launched.
CustomerManagedKey	*<Optional input>*	ARN of a customer managed KMS encryption key (CMK) to use for encryption and decryption of original data files during the ETL pipeline and query computation in AMC.
DataBucketName	`<Requires input>`	Name of the S3 bucket from which source data will be uploaded. Bucket is NOT created by this CloudForm ation.

1. Choose **Next**.
2. On the **Configure stack options** page, choose **Next** after reviewing the settings*.*
3. On the **Review and create** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
4. Choose **Submit** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a `CREATE_COMPLETE` status in approximately 10 minutes.

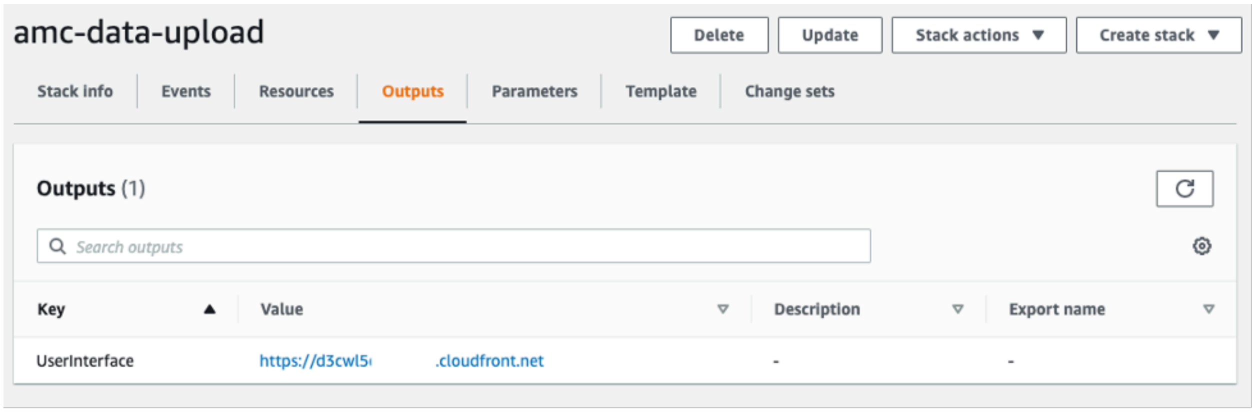
Step 4: Access the web interface

After the CloudFormation stack has been successfully deployed, navigate to the **Outputs** tab to retrieve the URL for the application. The solution sends a sign-in email containing information to access the web application, including web application URL, username, and a temporary password. The email is sent to the address that was specified in the **AdminEmail** parameter. The first time you log in to the application, you will be prompted to change your password.

Identify the web interface URL

1. Sign in to the AWS CloudFormation console and select the solution's stack.
2. On the **Stacks** page, select the stack.
3. Choose the **Outputs** tab.
4. Under the **Key** column, locate **UserInterface**, and select the corresponding value.
5. Open the web application in a new tab or browser window.
6. Sign in with your username (Admin email) and temporary password provided in the invitation email.
7. After signing in, follow the prompts to create a new password.

Web interface URL



Key ▲	Value	Description	Export name
UserInterface	https://d3cwl5 .cloudfront.net	-	-

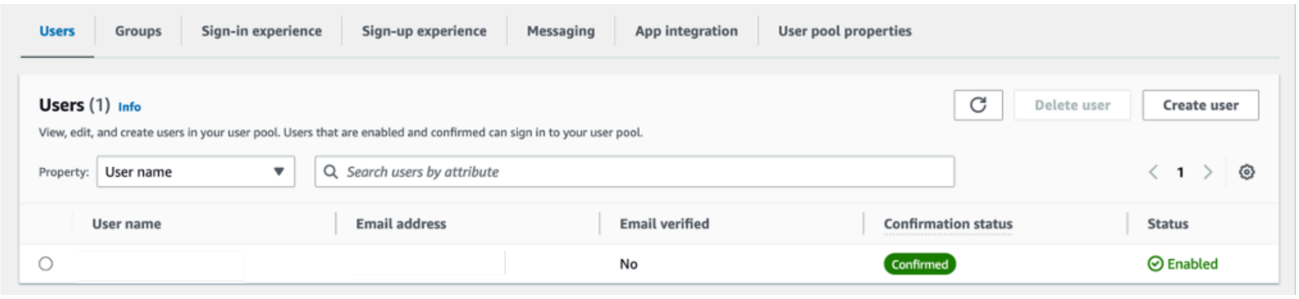
Step 5: (optional) Create user accounts

If more than one user needs access to the web application, the solution administrator can create additional users using the following procedure in Amazon Cognito.

1. Sign in to the Amazon CloudFormation console.

2. Open the AuthStack nested within your base CloudFormation stack.
3. Select the **Resources** tab, and select the **UserPool** resource.
4. On the **User pools** page, select the user pool.
5. On the **Users** tab, select **Create user**.

Amazon Cognito user pools create a user

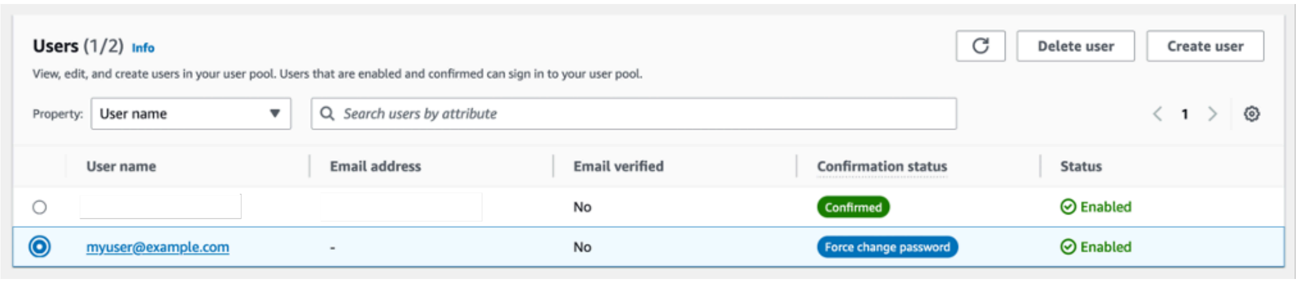


View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

	User name	Email address	Email verified	Confirmation status	Status
○			No	Confirmed	Enabled

1. In the **Create user** form, enter a user name and temporary password (ensure the options to send an invitation to the user and the verifications for phone number and email are not selected).
2. Select **Create user**.
3. On the **User pool** page, select the user you just created.

Amazon Cognito user pools page

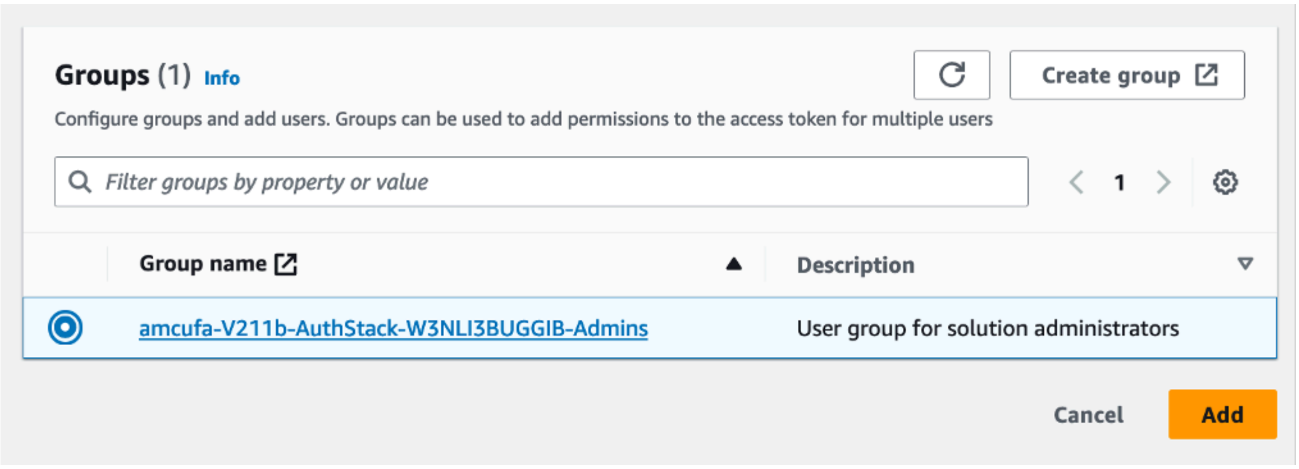


View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

	User name	Email address	Email verified	Confirmation status	Status
○			No	Confirmed	Enabled
●	myuser@example.com	-	No	Force change password	Enabled

1. On the **User** page, choose **Add user to a group**.
2. Select the group associated with the **AdminGroup** resource that is shown within the AuthStack nested stack in AWS CloudFormation.

Add a user to a group



Groups (1) Info

Configure groups and add users. Groups can be used to add permissions to the access token for multiple users

	Group name	Description
●	amcufa-V211b-AuthStack-W3NLI3BUGGIB-Admins	User group for solution administrators

The user can now access the web application, upload media files, and run the analysis workflows.

> **Note**
>
> (optional) We strongly encourage you to set up Multi-Factor Authentication (MFA) for each new user. For details, refer to Adding Multi-Factor Authentication (MFA) to a User Pool in the *Amazon Cognito Developer Guide*.

Monitor the solution with AppRegistry

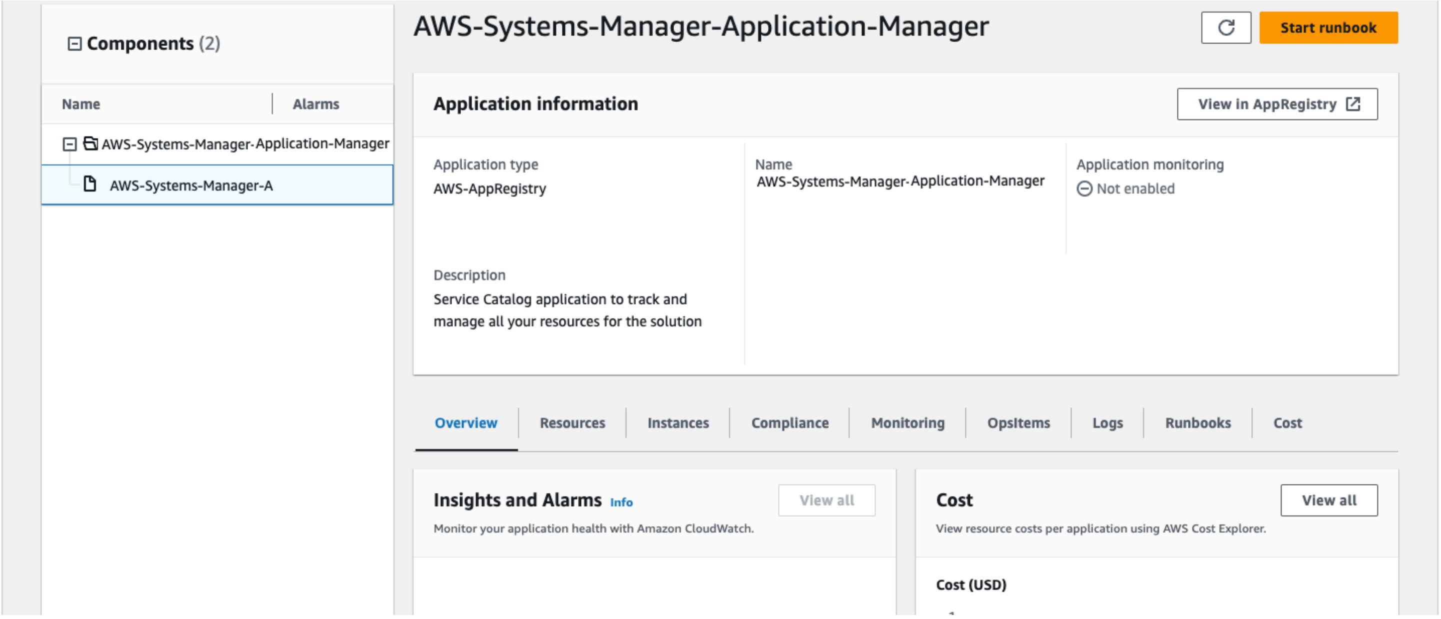
The solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager.

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution in the context of an application. For example, deployment status, CloudWatch alarms, resource configurations, and operational issues.

The following figure depicts an example of the application view for the solution stack in Application Manager.

Depicts solution stack in Application Manager



AWS-Systems-Manager-Application-Manager

Components (2)

Name	Alarms
AWS-Systems-Manager-Application-Manager	
AWS-Systems-Manager-A	

Application information

Application type
AWS-AppRegistry

Name
AWS-Systems-Manager-Application-Manager

Application monitoring
Not enabled

Description
Service Catalog application to track and manage all your resources for the solution

Insights and Alarms Info
Monitor your application health with Amazon CloudWatch.

Cost
View resource costs per application using AWS Cost Explorer.

Cost (USD)

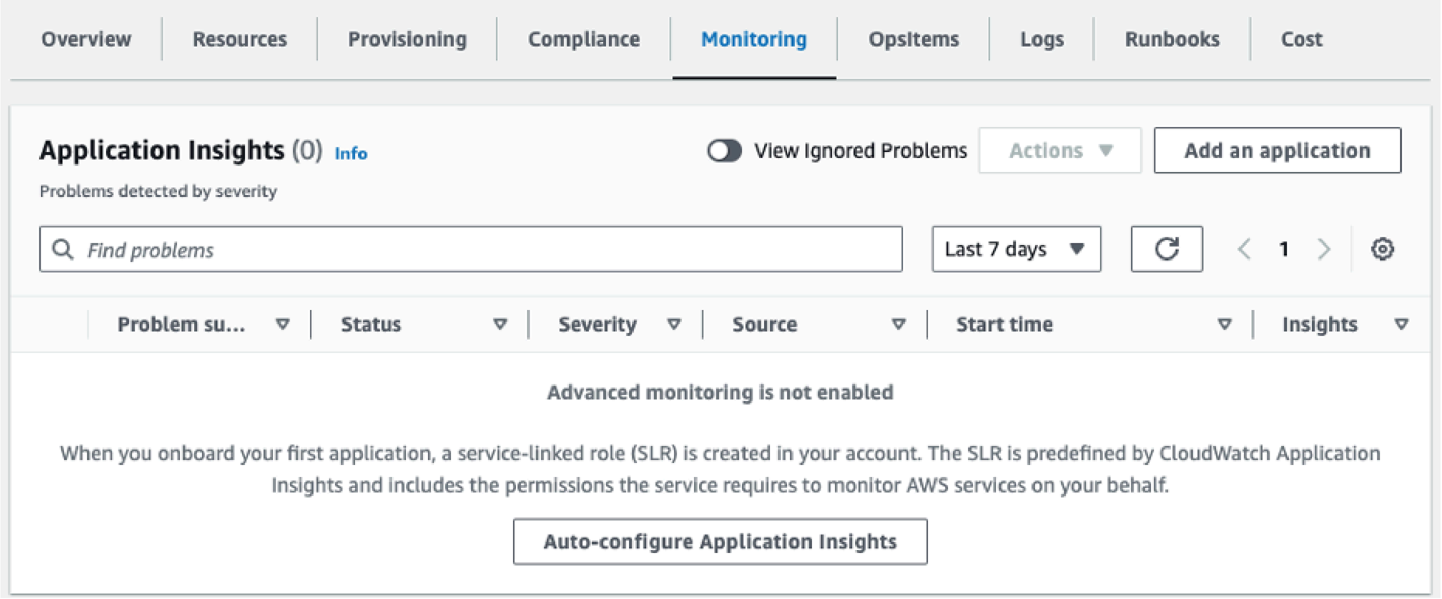
Activate CloudWatch Application Insights

1. Sign in to the Systems Manager console.

2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, search for the application name for this solution and select it.

   The application name will have **App Registry** in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.
4. In the **Components** tree, choose the application stack you want to activate.
5. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Insights**.

Application Insights dashboard showing no detected problems and option to auto-configure.

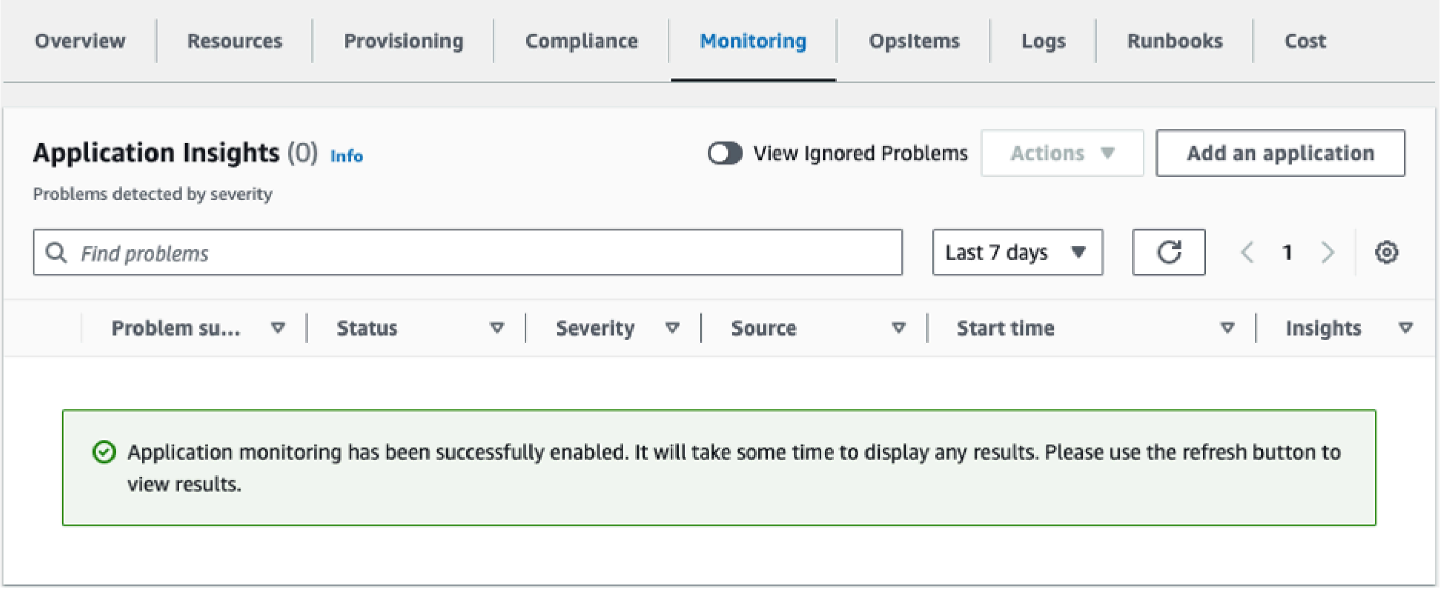


Advanced monitoring is not enabled

When you onboard your first application, a service-linked role (SLR) is created in your account. The SLR is predefined by CloudWatch Application Insights and includes the permissions the service requires to monitor AWS services on your behalf.

Monitoring for your applications is now activated and the following status box appears:

Application Insights dashboard showing successful monitoring activation message.



Application monitoring has been successfully enabled. It will take some time to display any results. Please use the refresh button to view results.

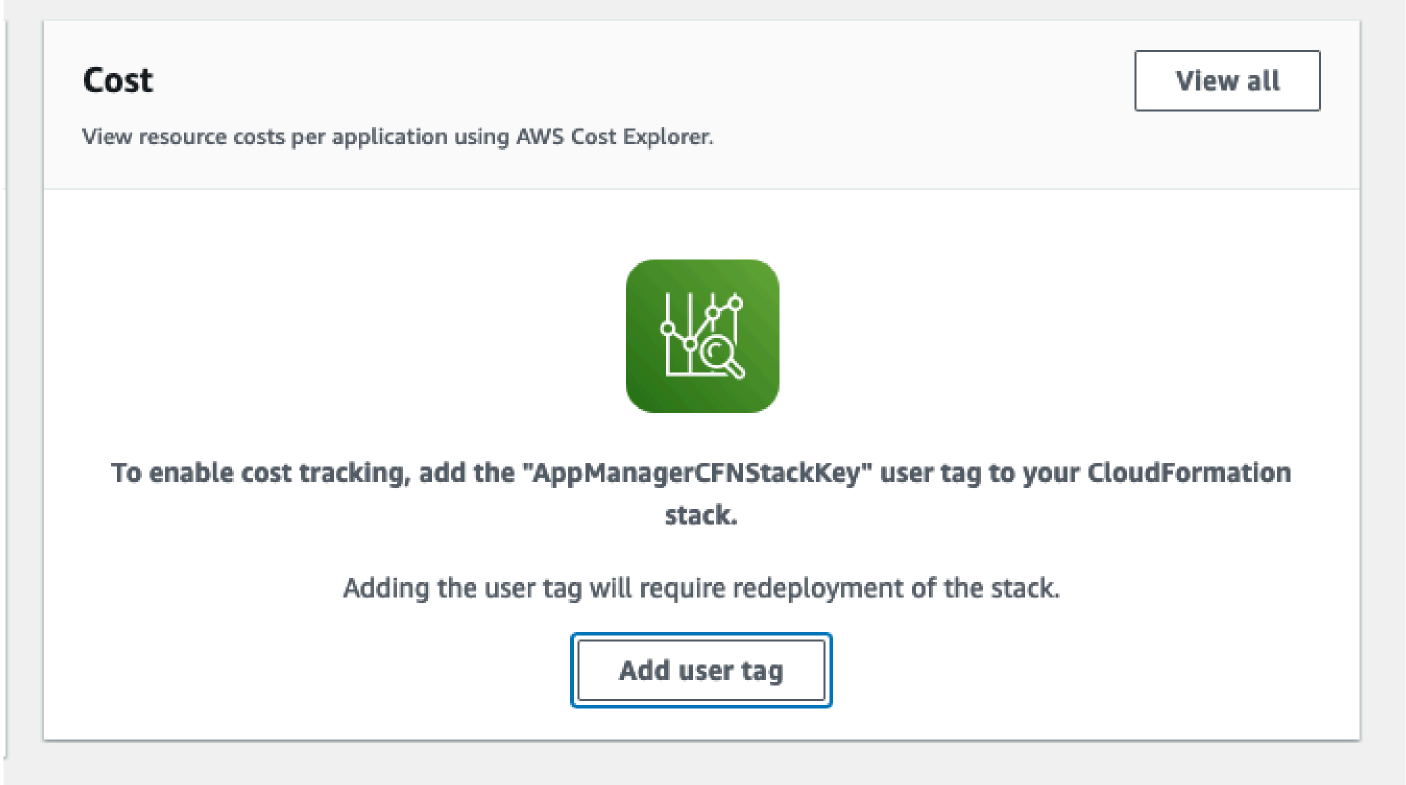
Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.

   The application name will have **App Registry** in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.

   Screenshot depicting the Application Cost add user tag screen



Cost

View resource costs per application using AWS Cost Explorer.

To enable cost tracking, add the "AppManagerCFNStackKey" user tag to your CloudFormation stack.

Adding the user tag will require redeployment of the stack.

5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate the cost allocation tags associated with this solution to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

1. Sign in to the AWS Billing and Cost Management and Cost Management console.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the AppManagerCFNStackKey tag, then select the tag from the results shown.
4. Choose **Activate**.

AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer, which must be first activated. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time. To activate Cost Explorer for the solution:

1. Sign in to the AWS Cost Management console.
2. In the navigation pane, select **Cost Explorer** to view the solution's costs and usage over time.

Update the solution

If you have previously deployed Amazon Marketing Cloud Uploader from AWS v3.0.0 or later, follow this procedure to update the Amazon Marketing Cloud Uploader from AWS CloudFormation stack to get the latest version of the solution’s framework.

> **Note**
>
> When you update a stack, the web interface will inherit all of the settings from the previous deployment; however, the CloudFront URL for accessing it will change.

1. Sign in to the CloudFormation console, select the base stack of your existing Amazon Marketing Cloud Uploader from AWS deployment, and choose **Update**.
2. Select **Replace current template**.
3. Under **Specify template**:
   a. Select **Amazon S3 URL**.
   b. Copy the link of the latest template.
   c. Paste the link in the **Amazon S3 URL** box.
   d. Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**.
4. Under **Parameters**, review the parameters for the template and modify them as necessary. For details about the parameters, refer to Step 3: Launch the stack in this guide.
5. Choose **Next**
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Select the box acknowledging that the template might create (IAM) resources.
8. Choose **Update stack** to deploy the stack.

   You can view the status of the stack in the CloudFormation console in the **Status** column. You should receive a `UPDATE_COMPLETE` status in approximately 15 minutes.

> **Note**
>
> To access the web interface, select the **Outputs** tab of the base CloudFormation stack, and use the URL specified for **UserInterface**.

Versions earlier than v3.0.0 cannot be upgraded.

Troubleshooting

If these instructions don't address your issue, see the Contact AWS Support section for instructions on opening an AWS Support case for this solution.

Problem: I have not received my temporary password to the web UI

Resolution

Logins can take up to 15 minutes to arrive. Look for an email from <no-reply@verificationemail.com>.

Problem: AWS Glue job has status FAILED

Resolution

If the AWS Glue job has failed, it might be due to selecting the wrong file format or that your file is not formatted correctly using the formats from the AMC data upload file format requirements.

To troubleshoot the error:

1. Sign in to the Amazon CloudWatch console.
2. Choose **Logs** from the left navigation pane.
3. Choose **Log groups**, and select the `/aws-glue/jobs/error` log group.
4. Look for more detailed information about the job failure in the latest log stream.

   If you see that the job failed with the error `Command failed with exit code 10` and you are trying to process files larger than 1GB, then set worker type under the Job details to `G 2X (8vCPU and 32GB RAM)` and rerun the job.

Problem: CloudFormation stack deployment fails

Resolution

The most common reason why the CloudFormation stack deployment fails is due to incorrect parameters. Ensure the following:

- The AWS account you are deploying this solution into matches the **Connected AWS Account** in the AMC UI.
- The Region you're using is the same Region that your AMC instance is deployed in.
- The AMC endpoint URL matches exactly what is set in your AMC instance.
- The AMC Data AWS account ID matches exactly what is set in your AMC instance.

Problem: Why did so few identities resolve for my dataset?

Resolution

AMC resolves identities by using the hashed PII fields in uploaded data. Advertisers must normalize those fields before hashing in a way that is consistent with how Amazon normalizes PII fields for the hashed identities it supplies to AMC. AMC resolves identities when the hashed PII in advertiser tables matches the hashed PII in Amazon tables. See ID resolution and supported identifiers document for more detail.

This solution attempts to normalize advertiser data in a way that is consistent with Amazon Ads, however it is possible for inconsistencies to be present. If you see poor identity resolution results for data that you uploaded using this solution, then use the AMC File Preparation Tool to generate hash files for your data as described in the *AMC Data Upload Documentation.pdf* document and upload those files to AMC using this solution. This approach will allow you to perform normalization using logic that more closely follows the normalization used by Amazon Ads. You can download the *AMC Data Upload Documentation.pdf* document from the Documentation link shown on your AMC instance administration page.

Contact Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can't resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

You can uninstall the Amazon Marketing Cloud Insights on AWS solution from the AWS Management Console or by using the AWS Command Line Interface (AWS CLI). You must manually delete the Amazon S3 buckets created by this solution. AWS Solutions implementations do not automatically delete Amazon S3 buckets in case you have stored data to retain.

Using the AWS Management Console

1. Sign in to the CloudFormation console.
2. On the **Stacks** page, select this solution’s installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to What is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name> --region <aws-region>
```

Use the solution

Allow a Return URL

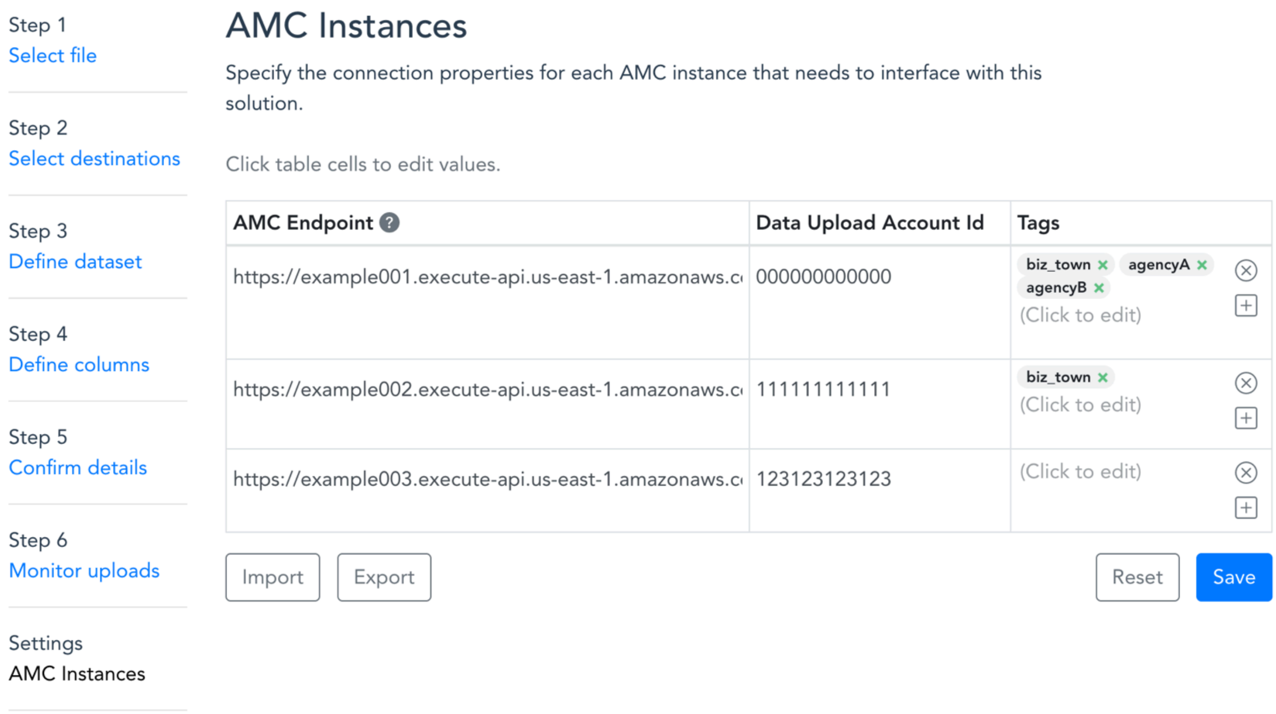
This solution requires an Amazon Ads developer account and Login with Amazon (LwA) application to authorize to Amazon Ads API. Before using this solution, you must set up an allowed return URL in LwA.

1. Copy the CloudFront URL from stack outputs.
2. See the Allow a return URL documentation to set the redirect_uri to the CloudFront URL/redirect. An example of the return URL of this solution is https:// `aaabbb123` .cloudfront.net/redirect.

Specify AMC instances

When signing in for the first time, you will be redirected to the **Settings** page to specify the OAuth credentials and the connection attributes for one or more AMC instances.

AMC Instances table with AMC Endpoint, Data Upload Account Id, and Tags columns for configuration.



AMC Instances

Specify the connection properties for each AMC instance that needs to interface with this solution.

Click table cells to edit values.

AMC Endpoint ?	Data Upload Account Id	Tags
https://example001.execute-api.us-east-1.amazonaws.c	000000000000	biz_town × agencyA × agencyB × (Click to edit)
https://example002.execute-api.us-east-1.amazonaws.c	111111111111	biz_town × (Click to edit)
https://example003.execute-api.us-east-1.amazonaws.c	123123123123	(Click to edit)

- **Client ID and Secret** - Credentials to access the Amazon Ads API.
- **AMC Instance Id** - Identifier of an AMC instance. This is available in the AMC account landing page.
- **Amazon Ads Advertiser ID** - The AMC account identifier that an AMC instance is linked to.
- **Amazon Ads MarketPlace ID** - The marketplace identifier for the marketplace in the request. The marketplaces are tied to the country.
- **Data Upload Account Id** - AWS account ID that is used to upload data to an AMC instance.
- **Tags** - Arbitrary strings can be saved as tags to help organize the AMC instance list. Tags can also be used to help find specific AMC instances from the instance selector dialog provided elsewhere in the UI.

To retrieve the OAuth credentials and AMC instance connection properties, see Retrieving Client ID and Client Secret and Locating AMC instance information.

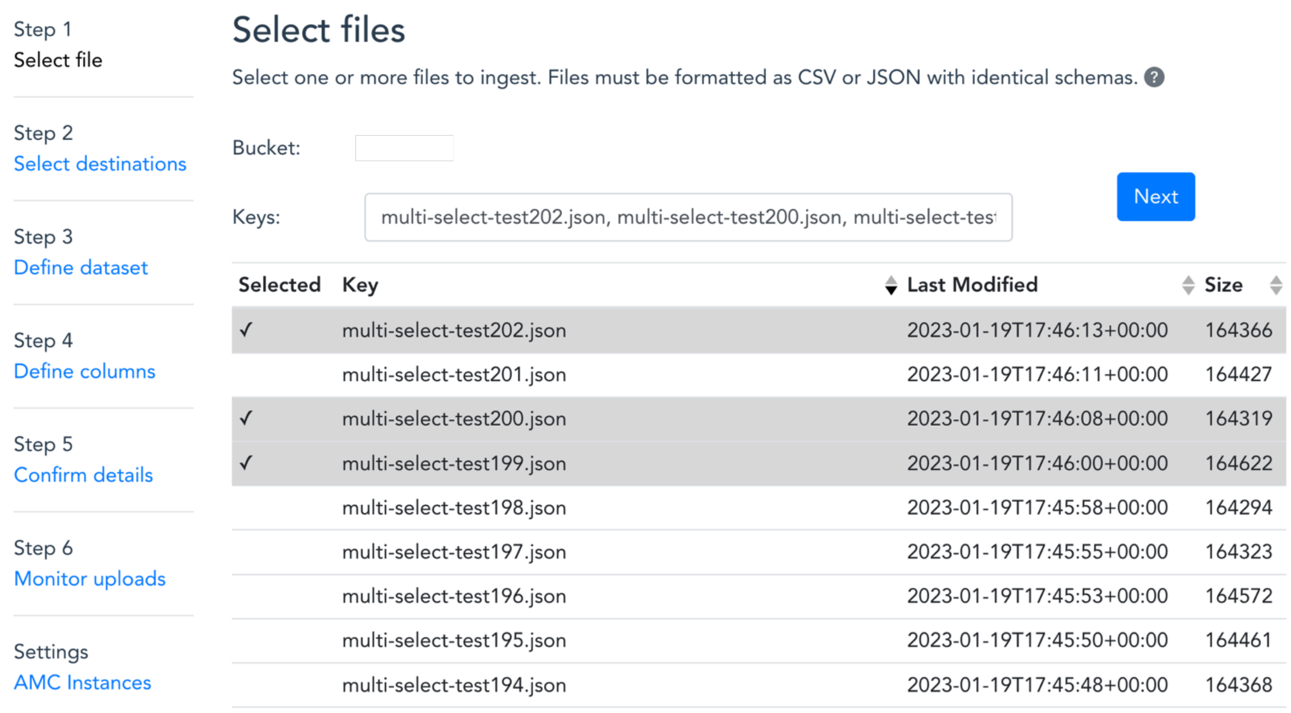
Select files

1. Select **Step 1 - Select file**. This displays all of the files that are available in the Amazon S3 bucket that you created in Step 1.

2. Select one or more files you want to use for the dataset.

3. Choose **Next**.

File selection interface showing multiple JSON files with their sizes and modification dates.



Select files

Select one or more files to ingest. Files must be formatted as CSV or JSON with identical schemas. ?

Bucket:

Keys: multi-select-test202.json, multi-select-test200.json, multi-select-tes

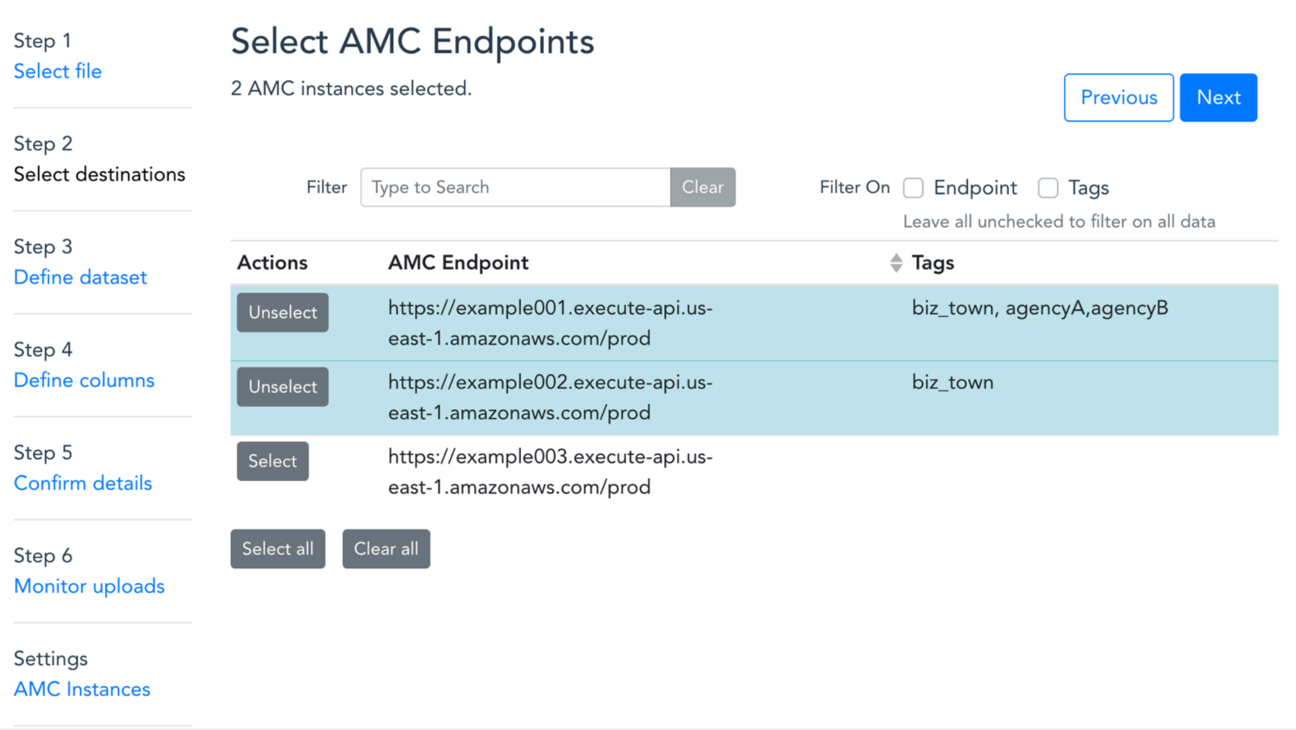
Selected	Key	Last Modified	Size
✓	multi-select-test202.json	2023-01-19T17:46:13+00:00	164366
	multi-select-test201.json	2023-01-19T17:46:11+00:00	164427
✓	multi-select-test200.json	2023-01-19T17:46:08+00:00	164319
✓	multi-select-test199.json	2023-01-19T17:46:00+00:00	164622
	multi-select-test198.json	2023-01-19T17:45:58+00:00	164294
	multi-select-test197.json	2023-01-19T17:45:55+00:00	164323
	multi-select-test196.json	2023-01-19T17:45:53+00:00	164572
	multi-select-test195.json	2023-01-19T17:45:50+00:00	164461
	multi-select-test194.json	2023-01-19T17:45:48+00:00	164368

Select AMC destinations

1. Select **Step 2 - Select destinations**. This displays all of the AMC instances that have been saved under the **Settings** page.

2. Select one or more AMC instances to receive this dataset. Use the search field to filter the AMC instance list by instance IDs and tags.

3. Choose **Next**.



Select AMC Endpoints

2 AMC instances selected.

Actions	AMC Endpoint	Tags
Unselect	https://example001.execute-api.us-east-1.amazonaws.com/prod	biz_town, agencyA,agencyB
Unselect	https://example002.execute-api.us-east-1.amazonaws.com/prod	biz_town
Select	https://example003.execute-api.us-east-1.amazonaws.com/prod	

Define the dataset

1. Select **Step 3 - Define dataset**.
2. To create a new dataset, enter a name for the dataset. This will be the table name that you query within AMC. This must be unique to your AMC instance.

Alternatively, to update an existing dataset, select **Add to existing dataset** and select the dataset from the provided drop-down menu.

1. (Optional) Enter a description. This will be used to detail what this dataset is to others who may not be familiar within the AMC instance.
2. Select the appropriate dataset period. When uploading time-series data, each file must be partitioned according to a specific unit of time. This unit of time is referred to as the dataset period.

The available periods are:

- **PT1H (hourly)**
- **P1D (daily)**

- **P1M (monthly)**

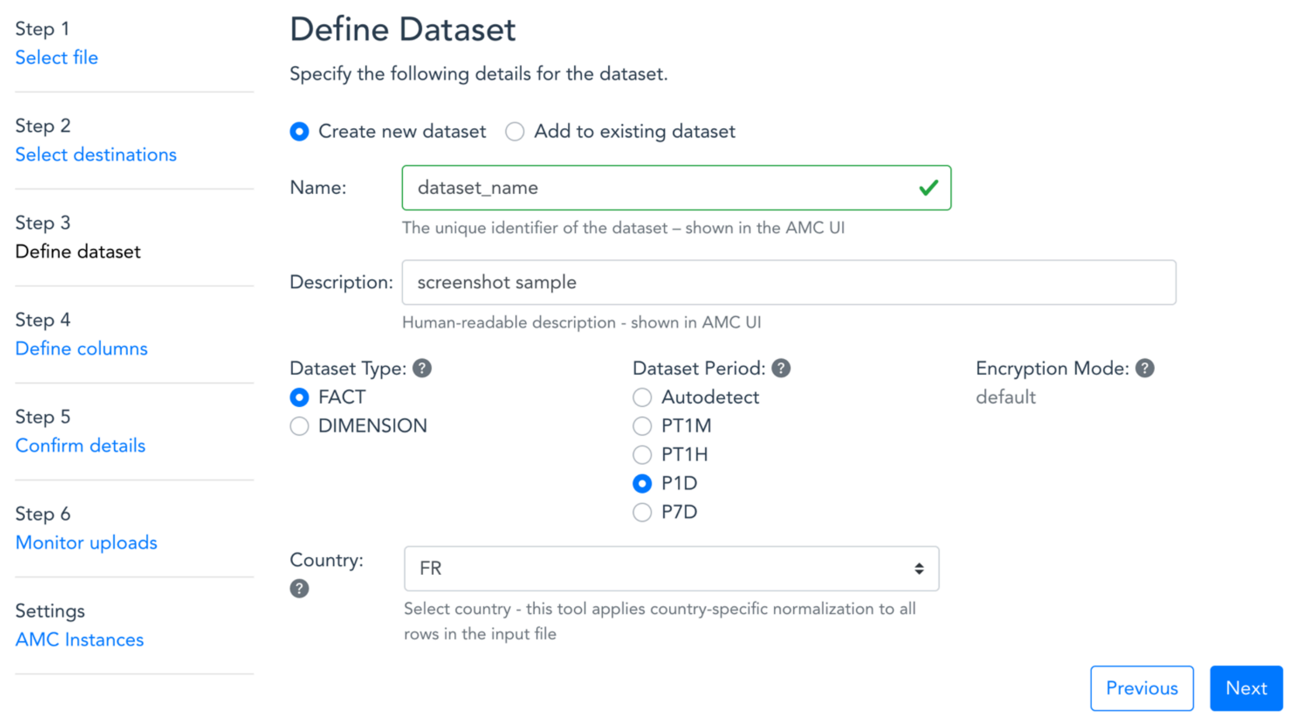
Select "DIMENSION" if no specific partition scheme is required for the dataset creation.

For details, refer to the AMC FACT vs. Dimension Datasets section of the Prepare Data document.

- **DIMENSION** - Dimension datasets can upload to a static table or to any information which is not time bound. Some examples include CRM audience lists, campaign metadata, mapping tables, and product metadata (such as a table mapping ASINs to external product names, or sensitive cost-of-goods-sold data). When uploading data to a dimension table, each upload is treated as a full replace - AMC queries will also use data from the last file uploaded.
- **FACT** - Fact datasets should be used for time-series data: data where each row has a corresponding date or timestamp associated. When defining a fact dataset, it is mandatory to designate one column as the main event time. Data must be segmented by day and must contain a Timestamp column.

  1. Select the appropriate country code for the data that you're preparing to upload. Identities will be resolved and addresses normalized according to the rules of this country. Be sure that each input file contains data for a single country and that this locale is the same for each file. For example, if you have data with both FR (French) and US (American) records, then these records should be split into different files and uploaded separately because this application will apply the same country-specific normalization rules for each file.
  2. Choose **Next**.

Dataset configuration form with fields for name, description, type, period, and country.



Define Dataset

Specify the following details for the dataset.

● Create new dataset ○ Add to existing dataset

Name: dataset_name ✓

The unique identifier of the dataset – shown in the AMC UI

Description: screenshot sample

Human-readable description - shown in AMC UI

Dataset Type: ?
● FACT
○ DIMENSION

Dataset Period: ?
○ Autodetect
○ PT1M
○ PT1H
● P1D
○ P7D

Encryption Mode: ?
default

Country: ? FR

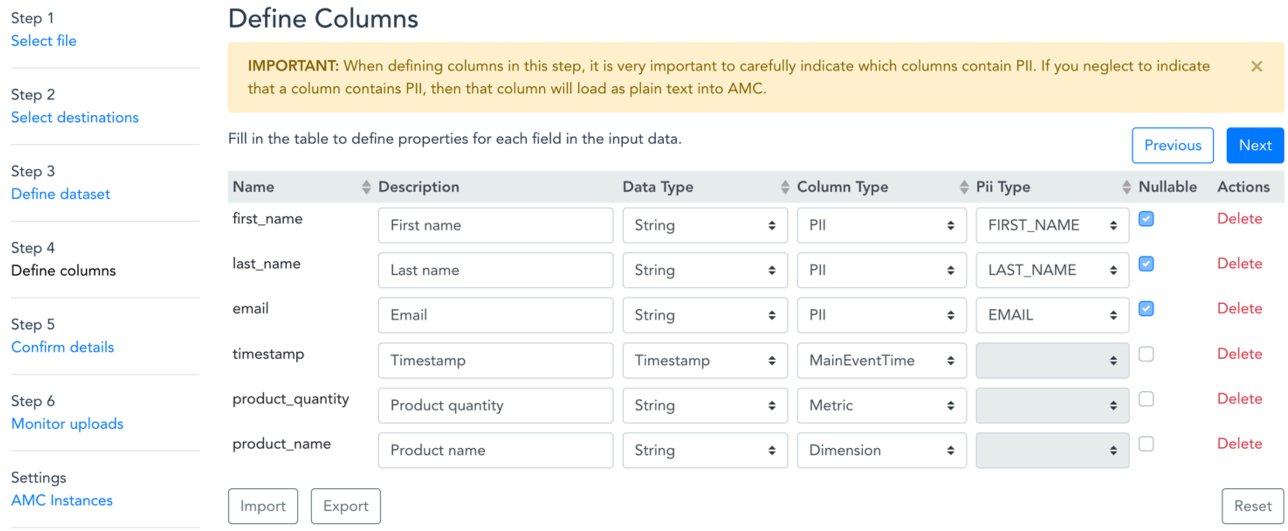
Select country - this tool applies country-specific normalization to all rows in the input file

Define the schema

1. Select **Step 4 - Define columns**.
2. Map the columns in your dataset to align with AMC's schema requirements.
3. Choose **Next**.

> **Important**
>
> When defining columns in this step, it is important to carefully indicate which columns contain PII. If you neglect to indicate that a column contains PII, then that column will not be obfuscated during the PII hashing phase of the AWS Glue job, and will subsequently load as plain text into AMC.

Define Columns interface showing fields for data import with PII indicators.



Step 1
Select file

Step 2
Select destinations

Step 3
Define dataset

Step 4
Define columns

Step 5
Confirm details

Step 6
Monitor uploads

Settings
AMC Instances

Define Columns

IMPORTANT: When defining columns in this step, it is very important to carefully indicate which columns contain PII. If you neglect to indicate that a column contains PII, then that column will load as plain text into AMC. ×

Fill in the table to define properties for each field in the input data.

Name	Description	Data Type	Column Type	Pii Type	Nullable	Actions
first_name	First name	String	PII	FIRST_NAME	☑	Delete
last_name	Last name	String	PII	LAST_NAME	☑	Delete
email	Email	String	PII	EMAIL	☑	Delete
timestamp	Timestamp	Timestamp	MainEventTime		□	Delete
product_quantity	Product quantity	String	Metric		□	Delete
product_name	Product name	String	Dimension		□	Delete

Column definitions:

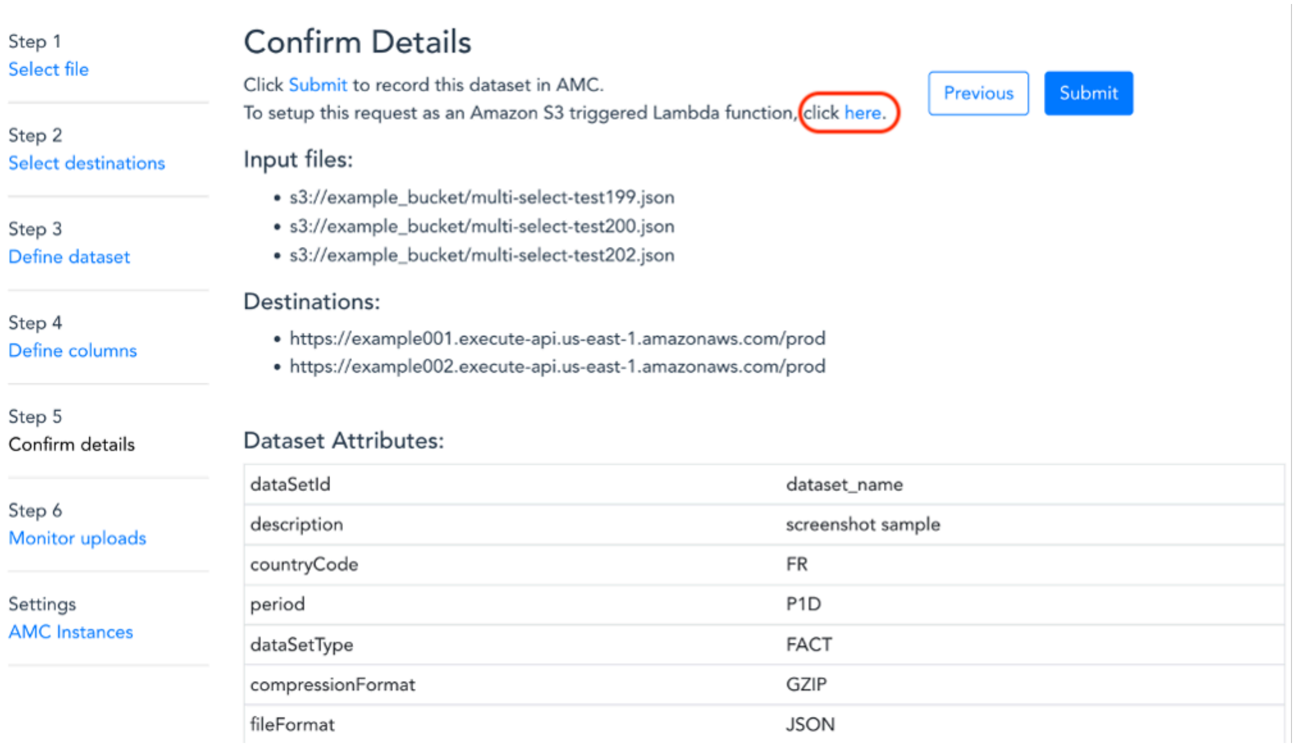
- **Data Type -** Select the data type that matches your column. This is relevant to the format of the data.
  - **String -** UTF-8 encoded character data
  - **Decimal -** Numerical with two floating point level precision
  - **Integer (32-bit) -** 32-bit numerical, no floating points
  - **Integer/LONG (64-bit) -** 64-bit numerical, no floating points
  - **TIMESTAMP -** Format: yyyy-MM-ddThh:mm:ssZ (ISO 8601)
  - **DATE -** Format: yyyy-MM-dd
- **Column Type -** Select the type of the column.
  - **PII -** A Personally Identifiable Information (PII) column contains sensitive information. Selecting PII requires you to define a PII Type to map the specific column to an identifier within Amazon Ads.
  - **Dimension -** These columns represent dimensional data such as Campaign Names, Product Names, Product IDs, etc. These columns must be grouped in AMC's output.
  - **Metric -** These columns represent values such as sales, clicks, etc. They can be aggregated in the output using AMC's supported aggregate functions. DIMENSION columns must be grouped in the output.
  - **MainEventTime - (Required for FACT Dataset Type) -** Only a single column may have this Column Type. This column contains the related Timestamp that is used to identify the date range of the dataset.

- **PII Type** - This selector allows you to select what type of PII data exists within the column. These are DIMENSION values that are always Nullable.
- **Nullable** - If there's a chance that this column may be empty in one of your rows, select the **Nullable** checkbox.
- **Actions** - If there is a specific column you do not want to send to Amazon Marketing Cloud, delete the column. If there is additional PII in your dataset that is not reflected in the **PII Type** field, delete it. It is not best practice to share unhashed PII data with Amazon.

Confirm details

Verify that the dataset attributes are correct, then choose **Submit**.

> **Note**
>
> You can automatically start this ETL process for files copied to a designated Amazon S3 location by using an Amazon S3 initiated Lambda function. For details about how to set this up, select the relevant link on the **Confirm Details** screen.



Confirm Details

Click Submit to record this dataset in AMC.
To setup this request as an Amazon S3 triggered Lambda function, click here.

Input files:

- s3://example_bucket/multi-select-test199.json
- s3://example_bucket/multi-select-test200.json
- s3://example_bucket/multi-select-test202.json

Destinations:

- https://example001.execute-api.us-east-1.amazonaws.com/prod
- https://example002.execute-api.us-east-1.amazonaws.com/prod

Dataset Attributes:

dataSetId	dataset_name
description	screenshot sample
countryCode	FR
period	P1D
dataSetType	FACT
compressionFormat	GZIP
fileFormat	JSON

Monitor job and verify dataset successfully uploaded

Your schema will be created within AMC and the AWS Glue job will be submitted to run asynchronously. As soon as the AWS Glue job completes the transformation, the application will notify AMC to upload the data from the ETL artifact Amazon S3 bucket to the AMC instance.

1. Select **Step 6 - Monitor uploads**.
2. Select an AMC instance from the AMC Instance selector.
3. From this page you can monitor dataset creation, upload status, and AWS Glue ETL jobs.

   Datasets - This table shows information about each dataset existing in the selected AMC instance.

   Uploads - This table shows uploads which have been performed for the selected dataset.

   AWS Glue Jobs - This table shows information about the AWS Glue ETL jobs which have run in response to upload requests performed by users of this application.

> **Note**
>
> AWS Glue ETL results older than three days will be automatically removed from the ETL artifact bucket.

Developer guide

This section addresses the source code, customization, troubleshooting, AMC instance information, AMC data type, date, and file format requirements for this solution.

Source code

Visit our GitHub repository to download the source files for this solution and to share your customizations with others.

Customization guide

If you, as a developer need to implement new data normalizations in the provided AWS Glue job to maximize identity resolution in AMC for your specific datasets, use the source code editor in the AWS Glue console.

Retrieving Client ID and Client Secret

1. Set up an Amazon Ads developer account and Login with Amazon (LwA) application.
2. Create a new security profile or use an existing profile.
3. From the security profile you are using, download the Client ID and Client Secret.

See the Create a LwA application documentation for more information.

Locating AMC instance information

1. Sign in to your Amazon Ads account.
2. Locate your AMC account by selecting the account list dropdown on the upper right and selecting your AMC account.

From the list of your AMC instances, select the **Instance Info** link for the instance where you would like data uploaded.

1. The AMC Instance ID is shown on the AMC account landing page. Alternatively, grab the instance id displayed in the URL of the Instance Info page. The code that is prefixed with amc is the instance id.
2. To retrieve Amazon Ads Advertiser ID, grab the entity identifier value that is displayed in the URL. The alphanumeric code that is prefixed with ENTITY is your account identifier. You can also use the GET operation of the `/amc/accounts` endpoint that returns a list of all the accounts your client ID has access to. An example of an account identifier is: ENTITY1AA1AA11AAA1.
3. Data upload AWS account ID is available on the Instance Info page.

OAuth Flow in Amazon Ads API

Advertisers can choose to delegate to approved client applications via an OAuth 2.0 flow, allowing these client applications to access their data and services. Client applications are Amazon Ads API callers administered by Login with Amazon (LwA) and approved by Amazon Ads for API access. For more information, consult with Amazon Ads API authorization documentation.

Both the advertiser and client application must participate to make a successful call to the Amazon Ads API. The advertiser much first grant authorization to the client application through a one-time authorization code created by LwA. This one-time authorization code is associated with both the advertiser and the client. Then the client application may use the one-time authorization code to retrieve an access token and refresh token from LwA. Client applications must include the resulting access token (`Authorization: Bearer <access token>`), along with its identifier (`Amazon-Advertising-API-ClientId`) and a profile identifier representing that advertiser's account in a specific marketplace (`Amazon-Advertising-API-Scope`) in the request header to access that specific advertiser's data and services.

AMC data upload file format requirements

CSV file requirements

CSV files must **be UTF-8 encoded** and comma delimited. In Microsoft excel, save the file as **CSV UTF- 8 (comma-delimited)** format. When CSV files are uploaded, AMC will automatically convert data to the corresponding column type. For example, if `12423.56` is contained in the CSV file and is mapped to a **DECIMAL** type column, AMC will coerce the string value contained in the CSV file to the appropriate column type.

JSON file requirements

JSON files must contain one object per row of data. Do not use JSON arrays. Following is an example of the accepted JSON format:

```
{"name": "Product A", "sku": 11352987, "quantity": 2, "pur_time":
 "2021-06-23T19:53:58Z"}
{"name": "Product B", "sku": 18467234, "quantity": 2, "pur_time":
 "2021-06-24T19:53:58Z"}
{"name": "Product C", "sku": 27264393, "quantity": 2, "pur_time":
 "2021-06-25T19:53:58Z"}
{"name": "Product A", "sku": 48572094, "quantity": 2, "pur_time":
 "2021-06-25T19:53:58Z"}
{"name": "Product B", "sku": 18278476, "quantity": 1, "pur_time":
 "2021-06-26T13:33:58Z"}
```

AMC data types, timestamp, and date formats

Dataset columns can be defined with the following data types. Pay close attention to the accepted formats for **TIMESTAMP** and **DATE** columns. If values in CSV / JSON data do not meet the accepted format, the upload might fail.

Ensure all values in CSV / JSON data confirm the specified data type and format before uploading. Where possible, string values will be coerced to the corresponding numerical type and vice-versa, but no guarantees are made on the casting process.

Data type	Format	Example
STRING	UTF-8 encoded character data	`My string data`
DECIMAL	Numerical with two floating point level precision	`123.45`
INTEGER (int 32-bit)	32-bit numerical, no floating points	`12345`
LONG (int 64-bit)	64-bit numerical, no floating points	`1233454565875646`

Data type	Format	Example
TIMESTAMP	yyyy-MM-ddThh:mm:ssZ	`2021-08-02T08:00:00Z`
DATE	yyyy-MM-dd	`8/2/2021`

AMC FACT compared with DIMENSION datasets

Before data can be uploaded, a dataset (table) must be created to store that data. AMC supports two types of tables (also referred to datasets): **FACT** and **DIMENSION**. It is important to understand when to use each, and the associated implications.

Most importantly, **FACT** datasets are used to store time-series data. The data files must be partitioned by a unit of time before uploading to AMC. The partition type (period) is specified on the dataset, and the options are `per minute`, `per hour`, `per day`, and `per week`. The partition type has an impact on how the data can be queried. For example, `per week` partitioned data cannot be queried at the daily level, and `per day` partitioned data cannot be queried at the hourly level.

Dataset type	Usage	Requires timestamp column	Requires partition scheme (Period)
FACT	Time series data	Yes	Yes
DIMENSION	Static tables	No	No

Fact datasets

Fact datasets should be used for time-series data - data where each row has a corresponding date or timestamp associated. When defining a **FACT** dataset, you must designate one column as the main event time.

Importantly, **FACT** datasets are used to store time-series data. The data files must be partitioned by a unit of time before uploading to AMC. The partition type (period) is specified on the dataset, and the options are `per minute`, `per hour`, `per day`, and `per week`. The partition type has an impact on how the data can be queried. For example, `per week` partitioned data cannot be queried at the daily level, and `per day` partitioned data cannot be queried at the hourly level.

When uploading data to a **FACT** dataset, each upload is performed according to a specific period of time - this is how AMC determines which data to include when performing queries.

Dimension datasets

Dimension datasets can be used to upload a static table, or any information which is not time-bound. Some examples include CRM audience lists, campaign metadata, mapping tables, and product metadata (such as a table mapping ASINs to external product names, or sensitive cost-of-goods-sold data). When uploading data to a dimension table, each upload is treated as a full replace - AMC queries will also use data from the last file uploaded.

Dimension datasets **do not** require a main event time column. Dimension datasets do not require uploaded files to be portioned. AMC will always use the most recent file uploaded (full-replace method of updating data).

Reference

This section includes information about an optional feature for collecting unique metrics for this solution and a list of builders who contributed to this solution, and the licensing notice.

- The *AMC Data Upload Documentation [Beta].pdf* document is referenced several times in this guide. You can download the document from the **Documentation** link shown on your AMC instance administration page.
- All documentation for AMC APIs on the Amazon Ads API is available at Advanced tools center.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products.

AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the `amazon-marketing-cloud-uploader-from-aws.template` AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

   ```
   AnonymizedData:
       SendAnonymizedData:
         Data: Yes
   ```

   to:

   ```
   AnonymizedData:
       SendAnonymizedData:
         Data: No
   ```

4. Sign in to the CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, specify template section, select **Upload a template file**.

7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Step 3: Launch the stack in the Deploy the solution section of this guide.

Contributors

- Ian Downard
- Immanuel George
- Andrew Marriott
- Cassidy Neal
- Mike Olson
- Jim Thario
- Yang Qin
- Thyag Ramachandran

Revisions

Publication date: *January 2023*

Visit the CHANGELOG.md in our GitHub repository to track version-specific improvements and fixes.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.