Implementation Guide

Amazon Marketing Cloud Insights on AWS

Amazon Marketing Cloud Insights on AWS: Implementation Guide

Table of Contents

Deploy this solution to easily analyze and improve advertising and marketing campaigns that run on Amazon Ads

Advertisers and agencies often ask how they can use Amazon Web Services (AWS) to improve performance and understanding of their campaigns running on Amazon Ads. For example, customers want to use AWS to query and analyze results from Amazon Ads' cloud-based data clean room, Amazon Marketing Cloud (AMC), that allows them to perform custom campaign analytics with strict safeguards around data privacy and security. Builders seeking to use the AMC API in combination with AWS need to allocate development resources to submit specialized queries to an AMC instance, load AMC output into an AWS account, prepare and catalog datasets for analytics, and connect data tools for Business Intelligence (BI). This also requires customers to provision infrastructure in their AWS account, including deployment and configuration of storage, data processing, ETL, and visualization.

Amazon Marketing Cloud Insights on AWS helps advertisers and agencies running campaigns on Amazon Ads to easily deploy AWS services to store, query, analyze, and visualize reporting from the AMC API, reducing development time from weeks to hours. The solution uses a ready-to-deploy code repository available on GitHub to automatically provision Amazon Simple Storage Service (Amazon S3) and AWS Glue, with services pre-configured to run queries in AMC and visualize reports. Analysts and developers with an active AMC instance can use the solution to run queries and monitor ongoing campaign performance across customer metrics and dimensions. As one example, an advertiser can use this solution to combine AMC API results from multiple brand campaigns inside AWS, and surface consumer segments that have a high propensity to purchase products, and prioritize high-performing segments for increased ad spending while reducing wasteful ad spending.

The intended audience for using this solution’s features and capabilities in their environment includes solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals.

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

If you want to . . .	Read . . .
Know the cost for running this solution.	Cost

If you want to . . .	Read . . .
The estimated cost for running this solution in the US East Region is USD $171.40 per month.	
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions are supported for this solution.	Supported AWS Regions
View or download the AWS CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template

Features and benefits

The solution provides the following features:

Reduce development time from weeks to hours

This solution deploys AWS services to store, query, analyze, and visualize reporting from the AMC API, reducing development time from weeks to hours.

Analyze campaign metrics

This solution enables analysts to run queries and monitor ongoing campaign performance across metrics such as reach and frequency, and dimensions including geographic area, audience segment, and device.

Combine campaign reports

Bring AMC API query results from multiple brand campaigns inside your AWS account across advertising channels such as video, audio, display, and sponsored ads to gain a holistic and in-depth understanding of the customer journey.

Integration with Service Catalog AppRegistry and AWS Systems Manager Application Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager. With this integration, you can centrally manage the solution's resources.

Use cases

Monitor Amazon Ads campaign performance

This Solution helps advertisers and agencies running campaigns on Amazon Ads to easily deploy AWS services to store, query, analyze, and visualize reporting from the Amazon Marketing Cloud API, reducing development time from weeks to hours.

Concepts and definitions

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

application

A logical group of AWS resources that you want to operate as a unit.

Amazon Marketing Cloud (AMC)

Amazon Ads' cloud-based data clean room that allows advertisers to perform custom campaign analytics with strict safeguards around data privacy and security.

customer

This solution uses the term *customer* to refer to AMC instances onboarded through the Tenant Provisioning Service microservice. A customer has a 1-to-1 relationship with an AMC instance.

microservice

AWS resources deployed to facilitate managing AMC Instances, sending workflow requests to AMC API, and retrieving reports from Amazon Ads and Selling Partner Reporting API.

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

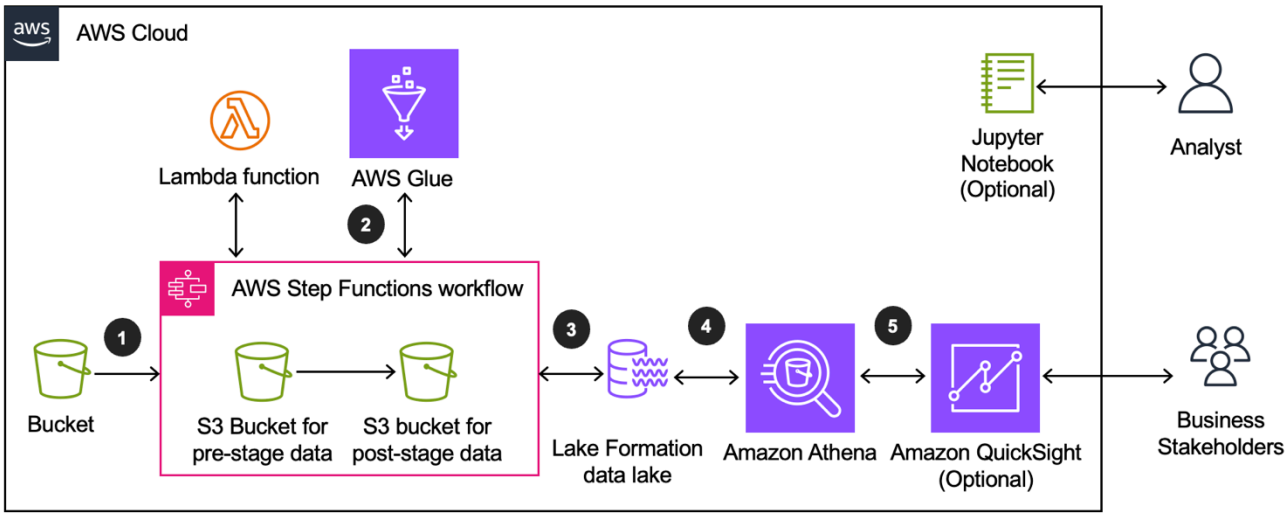
Architecture overview

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.

Depicts Amazon Marketing Cloud Insights on AWS architecture



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

1. Data that lands in a customer's onboarded AMC Amazon S3 bucket is picked up by the solution data lake pipeline and moved under the pre-stage S3 prefix in the stage S3 bucket.
2. AWS Glue applies transformation logic on the incoming data to prepare it for analysis, storing the result under post-stage S3 prefix in the stage S3 bucket.
3. AWS Lake Formation controls access permissions to the transformed data in the stage S3 bucket.
4. Users can access the transformed data using Amazon Athena to run query analysis on the stage S3 bucket.
5. Users can build dashboards in Amazon QuickSight from Athena queries run on the stage data.

> **Note**
>
> AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

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.

Operational excellence

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

Perform operations as code - This solution's infrastructure and workflows are entirely specified using CDK v2.0 in Python 3.x and deployed as a CloudFormation template. The entire solution can be deployed into a new AWS account without any other preparation, upgraded in place with a new version of the template, and removed from an account with a single command.

Make frequent, small, reversible changes - This solution is designed to be customized by the end-user, if desired. The solution can be forked from the GitHub repository into a customer's account, customized, rebuilt, hosted in a customer's Amazon S3 buckets, and deployed via CloudFormation. This process can be repeated iteratively to test changes to the default solution.

Security

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

Implement a strong identity foundation - All interactions among resources created by the solution are secured using AWS Identity and Access Management (IAM) roles, policies, and signature V4 request signing. All credentials used to interact among resources are temporary and typically have a lifetime of less than one hour.

Maintain traceability - Runtime logging by Lambda functions installed by the solution is sent to Amazon CloudWatch Logs and preserved with the default retention settings.

Apply security at all layers - Interactions among resources require permissions defined in the related resource's IAM role. No publicly accessible resources are created by this solution.

Protect data in transit and at rest - All data is encrypted in transit via TLS-protected API requests. All persistent resources are configured for encryption at rest. Several KMS keys are created in the customer's account for use in encryption and decryption with Amazon S3 and AWS Glue resources if needed.

Reliability

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

Automatically recover from failure - CloudWatch metrics and alarms are used to monitor the operation of the solution with the ability to notify users or other systems when thresholds are breached. Dead-letter queues (DLQs) are used to receive messages from Amazon Simple Queue Service (Amazon SQS) that cannot be processed due to a problem. Alarms on DLQs notify when a message is added to the DLQ.

Improved capacity planning - You can create alarms for error and throttle conditions with all Lambda functions, and these alarms can be monitored by users or external systems.

Manage change in automation - This solution's infrastructure and workflows are entirely specified using CDK v2.0 in Python 3.x and deployed as a CloudFormation template. This solution is designed to be customized by the end-user, if required. You can deploy and upgrade stacks, either manually or with automation systems for testing when new builds of a customized template are available. CloudFormation treats an upgrade as a transaction and can roll back a failed stack installation and restore the previous version automatically.

Performance efficiency

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

Go global in minutes - The CloudFormation template can be used to create a stack in any Region, and multiple stacks can co-exist in the same Region if needed for testing and production, for example.

Use serverless architectures - Lambda functions are the primary compute mechanism used throughout the solution, meaning compute will automatically scale as needed to handle the request load.

Cost optimization

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

Analyze and attribute expenditure - This solution is configured with AppRegistry, which supports accumulating cost data for each instance of the stack. Over time, you can see the impact of each stack deployment on your monthly account charges.

Sustainability

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

Anticipate and adopt new, more efficient hardware and software offerings - This solution utilizes the AWS Graviton2 processor for Lambda functions.

Use managed services - This solution is designed using serverless and AWS Managed Services. Most of the lower-level operational costs of maintaining data center hardware, operating systems, and starting and stopping infrequently used services is the responsibility of AWS. The cost of operating and maintaining applications and access by users is the responsibility of the customer.

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.

AWS services in this solution

AWS service	Description
Amazon Athena	**Core**. Access the AWS Glue Data Catalog and query the transformed data in the stage Amazon S3 bucket.
AWS Glue	**Core**. Apply a heavy transformation in the data lake including partitioning pre-stage data and output the data into parquet files.
AWS Lambda	**Core**. Lambda is used to add AMC instances as a part of microservices and register provision ed customers for the data lake. Lambda is also used to process workflow requests, check responses, notify users, transform raw data, partition pre-stage data, and manage metadata stored in Amazon S3 files.
AWS Lake Formation	**Core**. For data lake governance and security.
Amazon S3	**Core**. The solution uses Amazon S3 buckets to store reporting from Amazon Ads API and Selling Partner API, pre-stage data, and post-stage data.
AWS Step Functions	**Core**. Step Functions orchestrates the Lambda functions and user notifications in the Tenant Provisioning Service, Workflow Manager and data lake.

AWS service	Description
Amazon DynamoDB	**Supporting**. DynamoDB tables store details of tenants, workflows, and data lake transform ations.
Amazon EventBridge	**Supporting**. EventBridge captures the raw data landing into Amazon S3 buckets and invokes the data lake on a recurring basis.
AWS KMS	**Supporting**. The solution uses KMS keys to encrypt and decrypt the data in Amazon S3 buckets, SQS queues, and DynamoDB tables.
Amazon SNS	**Supporting**. The solution uses Amazon SNS to publish execution status of workflow management service.
Amazon SQS	**Supporting**. The solution uses Amazon SQS to send, store, and receive messages between tenants, workflows, and the data lake.
AWS Systems Manager	**Supporting**. Provides application-level resource monitoring and visualization of resource operations and cost data.
AWS Secrets Manager	**Supporting**. Secrets Manager stores the user-specified OAuth credentials.
Amazon QuickSight	**Optional**. For business intelligence, analytics, interactive dashboards, and visualizations that business stakeholders can use.
Amazon SageMaker Jupyter notebook	**ptional**. Amazon SageMaker AI with sample Jupyter notebooks that analysts can use to provision tenants and manage workflows.

Microservices

This solution deploys six microservices: Platform Management Notebooks, Tenant Provisioning Service, Workflow Manager, Amazon Ads Reporting, Selling Partner Reporting, and the Serverless Data Lake.

Platform Management Notebooks

The Platform Management Notebooks serve as sample code for interfacing with the Tenant Provisioning Service, Workflow Manager, Amazon Ads Reporting, and Selling Partner Reporting microservices.

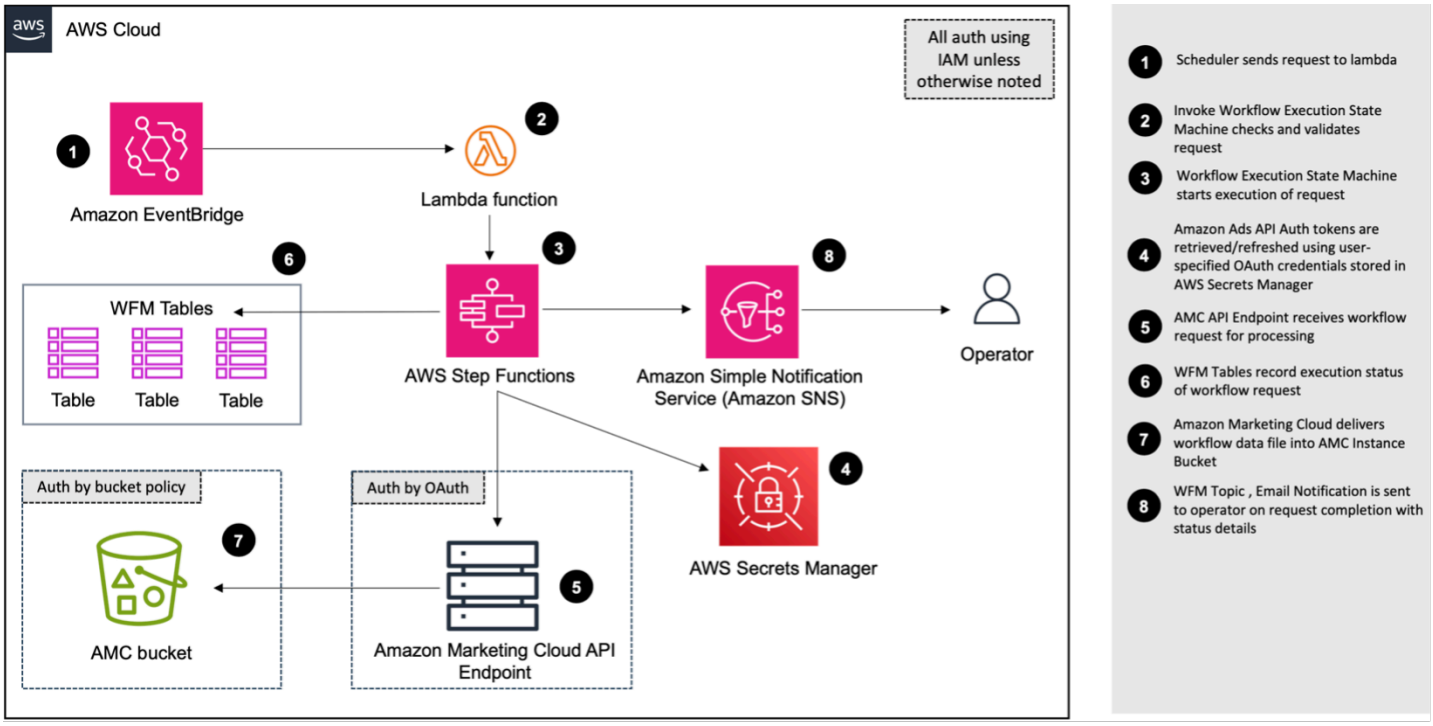
Tenant Provisioning Service

The Tenant Provisioning Service manages AMC customers onboarded through the solution. Each onboarded AMC customer is mapped to an AMC instance and deployed as a stack in the solution.

Workflow Manager

The Workflow Manager manages requests sent to the AMC API. In addition to synchronizing data between the solution and a customer's AMC instance, the Workflow Manager enables scheduling of AMC workflows using CRON-based scheduling, and queue-based routing to ensure that all requests are processed.

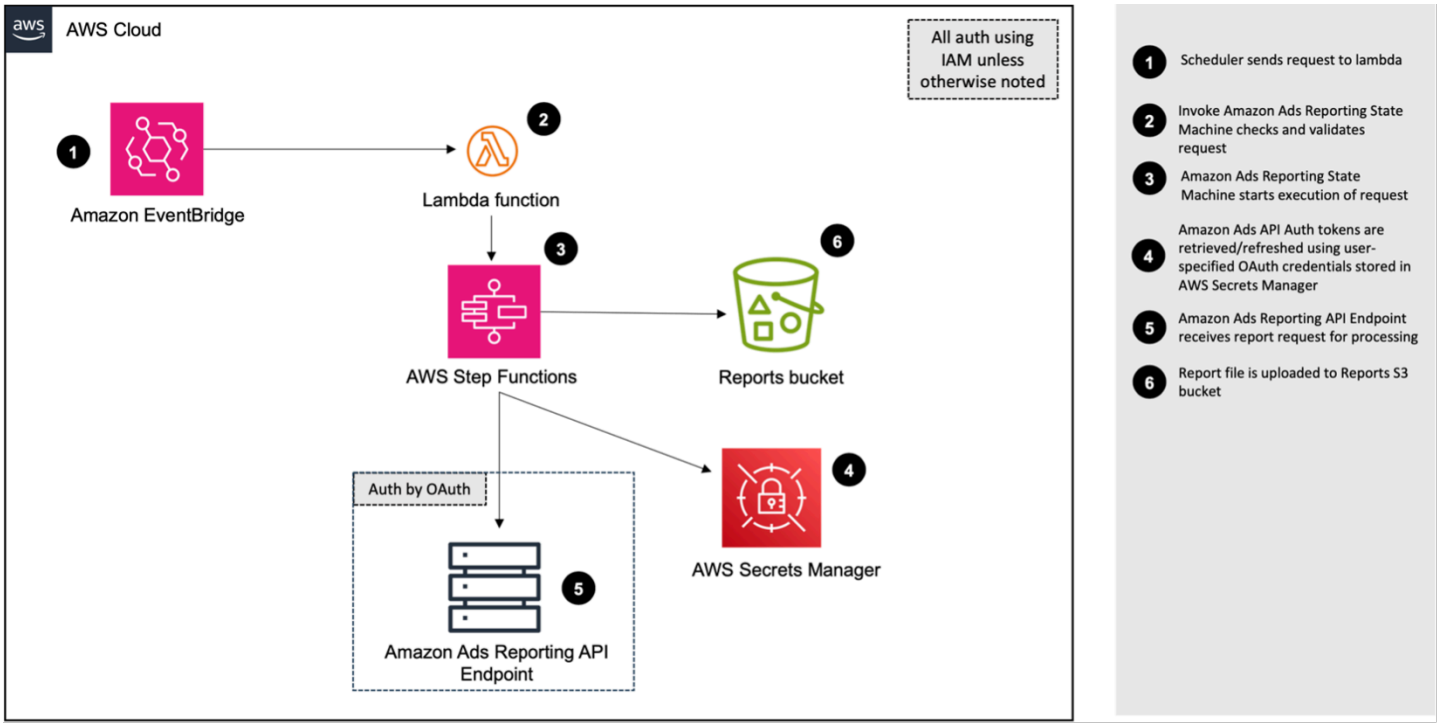
Depicts Workflow Manager



Amazon Ads Reporting

The Amazon Ads Reporting microservice schedules and fetches reports from the Amazon Ads reporting API endpoint.

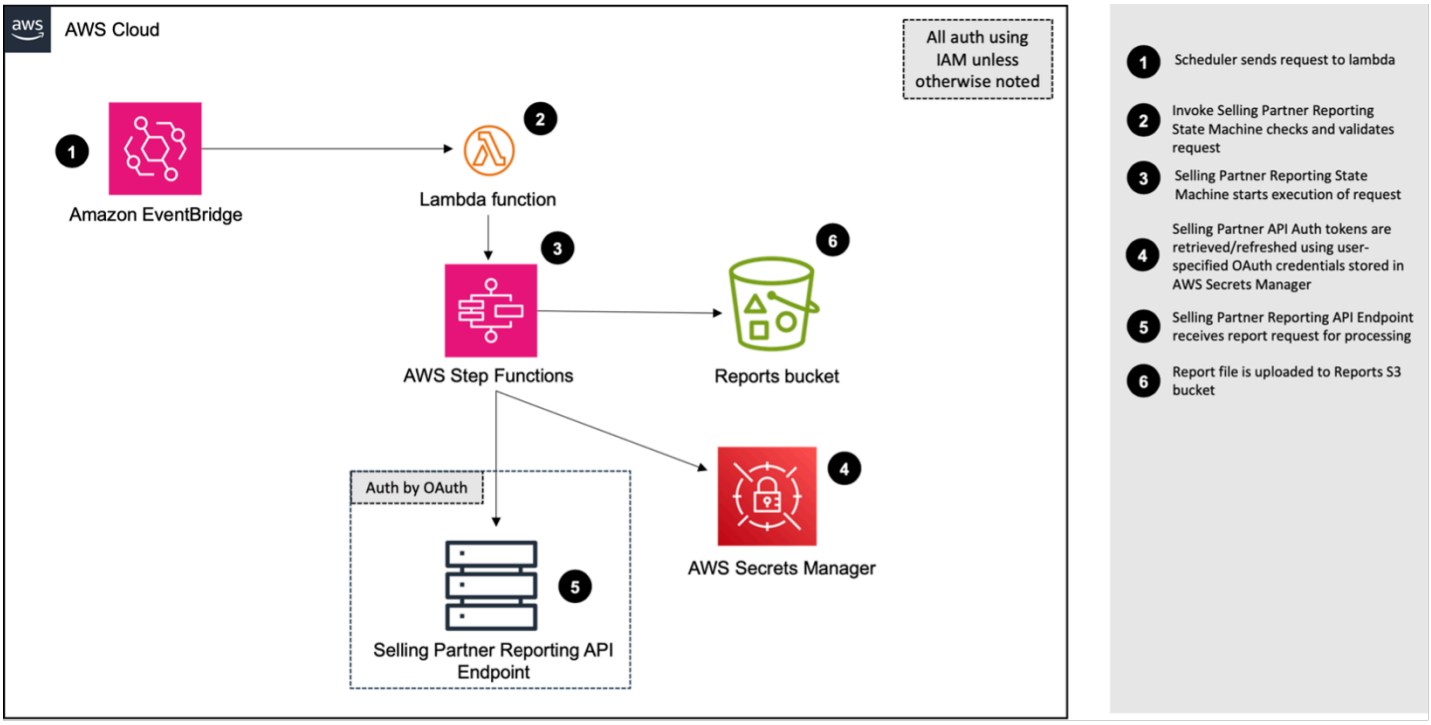
Depicts Amazon Ads Reporting



Selling Partner Reporting

The Selling Partner Reporting microservice schedules and fetches reports from the Selling Partner API.

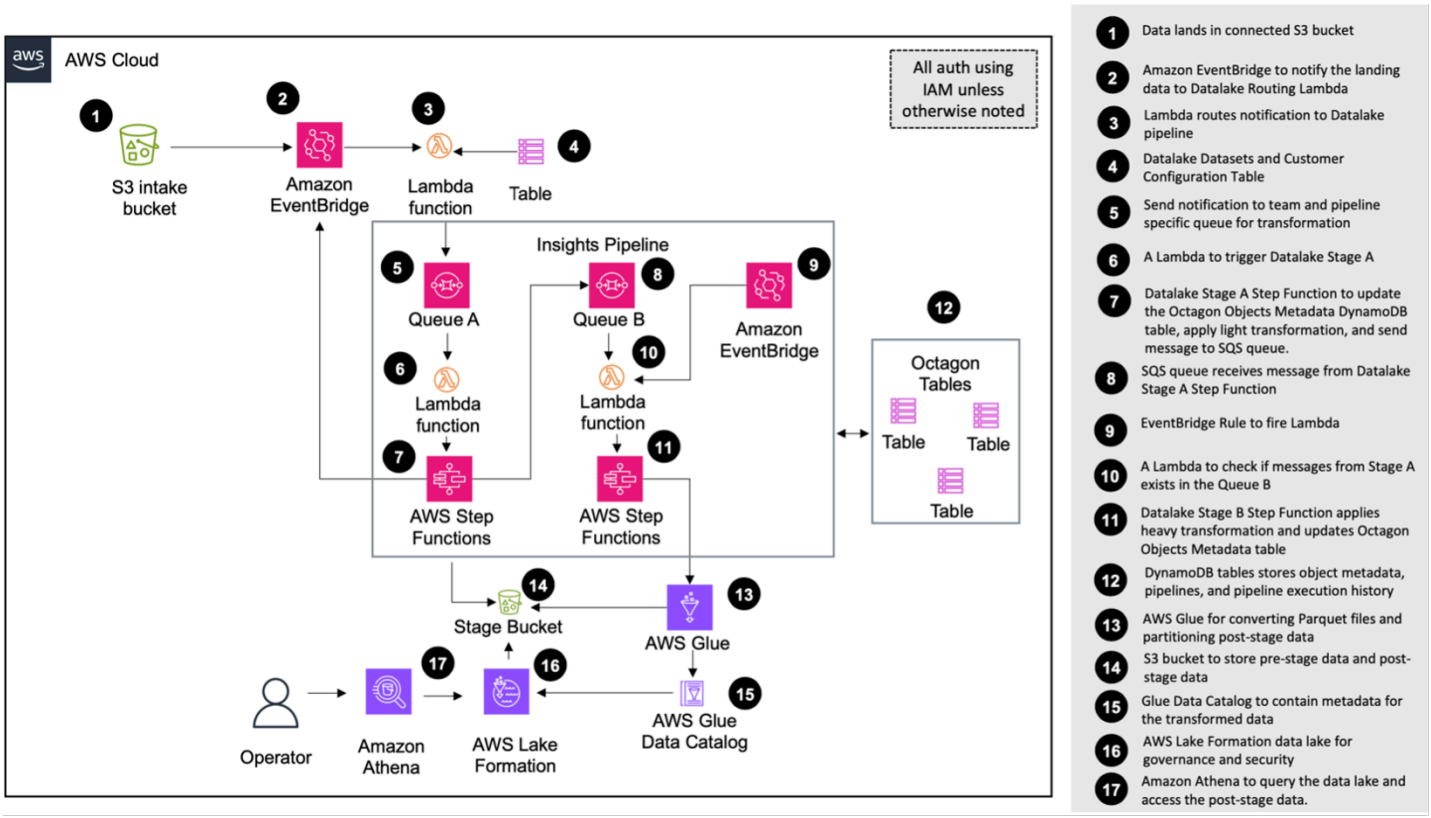
Depicts Amazon Ads Reporting



Serverless Data lake

The Data Lake transforms the data delivered by the other microservices in any of the intake S3 buckets deployed by the application (reporting bucket for Amazon Ads and Selling Partner reports, AMC buckets for AMC data, and the general-purpose Raw bucket for custom data uploaded by an external provider or AWS service). The data lake detects the objects created in the bucket and starts the transformations if the dataset is configured. The data lake routes the data to its corresponding pipeline and applies custom transformation for the dataset provided by customers. The transformed data is stored to the Amazon S3 stage buckets and can be accessed through AWS Glue Data Catalog.

Depicts Data Lake



Orchestration

AWS Step Functions is the orchestration service used in the Tenant Provisioning Service, Workflow Manager, Amazon Ads Reporting, Selling Partner Reporting, and data lake to coordinate multiple activities in this solution.

- The Step Functions in the Tenant Provisioning Service orchestrate Lambda functions to add AMC instances, and register the provisioned customer into the data lake.
- The Workflow Manager uses Step Functions to coordinate Lambda functions for processing workflow requests, creating workflow runs, checking workflow status, and notifying the user.
- Step Functions in the data lake automates transformations after data are delivered in any of the intake S3 buckets.
- The Amazon Ads Reporting and Selling Partner Reporting Step Functions orchestrate the Lambda functions to schedule and handle report requests, check the status of reports, and download the completed reports to the S3 bucket.

Plan your deployment

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

Supported AWS Regions

Amazon Marketing Cloud Insights on AWS is supported in the following AWS Regions:

Region ID	Region name
us-east-2	US East (Ohio)
us-east-1	US East (N. Virginia)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
ca-central-1	Canada (Central)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-northeast-3	Asia Pacific (Osaka)
eu-central-1	Europe (Frankfurt)
eu-west-1	Europe (Ireland)
eu-west-2	Europe (London)
eu-west-3	Europe (Paris)

Region ID	Region name
sa-east-1	South America (São Paulo)

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the cost for running this solution with the default settings in the default US East (N. Virginia) Region is approximately **$171.40 per month**.

We recommend creating a budget 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.

Sample cost table

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
Amazon S3	50GB S3 standard storage, 20,000 PUT requests, 1,000 GET requests.	$1.26
Amazon Athena	150 queries per day. 50 GB scanned per day.	$36.62
Amazon Lambda	150000 requests, 1024 MB memory, ARM architecture.	$0.00
Amazon Glue	2 DPUs for Apache Spark job, 54 hours duration for which Apache Spark ETL job runs, 1000 Data Catalog objects stored per month, 1000 Data Catalog access requests per month.	$47.53

AWS service	Dimensions	Cost [USD] / month
AWS Step Functions	1 tenant on boarded per month (12 state transitio ns per workflow), 1 WFM workflow created per month (12 state transitions per workflow), 2160 WFM workflow executions per month (80 state transitions per workflow), 4320 Data Lake stage A requests per month (7 state transitions per workflow), 2160 Data Lake stage B per month (20 state transitions per workflow).	$5.86
Amazon QuickSight (Optional)	5 readers, 22 working days per month: * Percent of active readers = 5 * Percent of frequent readers = 50 * Percent of occasiona l reader = 25 * Percent of inactive readers = 20 1 author	$31.40
Amazon Sagemaker (Optional)	Storage (General Purpose SSD (gp2)), Instance name (ml.t2.medium), Number of data scientist(s) (1), Number of On-Demand Notebook instances per data scientist (1), On-Demand Notebook hour(s) per day (4), On-Demand Notebook day(s) per month (22).	$4.36

AWS service	Dimensions	Cost [USD] / month
AWS Key Management Service	Number of customer managed Customer Master Keys (15), Number of symmetric requests (20000).	$15.06
Amazon DynamoDB	DynamoDB on-demand capacity, Table class (Standard), Average item size (all attributes) (1 KB), 1000000 on-demand writes per month, 1000000 on-demand reads per month.	$1.38
Amazon CloudWatch	Number of Metrics (26), Standard Logs: 40 GB data ingested.	$27.98
	Total	**$171.40**

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility 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

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 Lambda functions access to create Regional resources.

Secrets management

This solution leverages AWS Secrets Manager to securely store user-specified OAuth credentials and tokens. The solution generates the secrets using a JSON text string with predefined key-value pairs. The secrets are identified by a unique key name. The solution restricts access to this secret through IAM policies, allowing only specific Lambda functions that need it for operation to access or update this secret.

Restricted Amazon SageMaker permissions

The Amazon SageMaker instance has limited permissions: to only access the sample notebooks from the deployed artifacts S3 bucket; and to invoke the Lambda functions needed to use the microservices. The SageMaker instance does not have access to data contained within the solution or the Secret Manager. The notebooks send requests to Lambda using Boto3, with IAM policies restricting its functionality to invoke these functions only. The notebooks are optional and only serve as examples of how to use Boto3 to invoke the Lambda functions.

Amazon Ads authorization process

This solution provides a notebook and a Lambda function to facilitate users in the Amazon Ads authorization process. After obtaining an authorization code from Login with Amazon (LwA), users can input their client ID, client secret, and authorization code into Secrets Manager.

The Lambda function is invoked, which retrieves access and refresh tokens and stores them in Secrets Manager for future API calls. There is no input required from the user to invoke the Lambda function as the required values are stored in Secrets Manager ahead of time. This Lambda function has restricted permission and can only update the specific secret created by this solution.

Security recommendations

Create admin roles

We recommend that the admin create IAM roles and policies to control other users' access to the AWS resources created by this solution. Each user must have only the minimum permissions required to perform specific job functions. For more information, see Access management for AWS resources.

Rotate secrets

This solution uses Secrets Manager to store users' OAuth2 credentials, authorization code, access token, and refresh token. OAuth2 credentials are associated with the security profile created in LwA. Access tokens are valid for sixty minutes, and can be refreshed using the refresh token. The refresh token remains valid until the user who granted authorization revokes it.

Selling Partner API requires OAuth2 credential rotation every 180 days. See the Selling Partner API documentation for instructions on rotating your application's credentials. Therefore, we recommend rotating the OAuth2 credentials and refresh token based on their enterprises' password rotation policy. See Rotate AWS Secrets Manager secrets.

What to do if your tokens are compromised?

An LWA refresh token is a long-lived token. Generating a new refresh token does not invalidate previous refresh tokens. Therefore, if the tokens are compromised, the impact must be analyzed with the corresponding advertiser and advertiser client. We recommend that users contact LwA to get tailored recommendations for their specific scenario.

> **Note**
>
> See the LwA page for more information about access tokens and refresh tokens.

If you suspect that your tokens have been compromised, you must take the following actions, though these are not exhaustive:

1. Invalidate the tokens and delete the security grant from LwA.
2. Delete the entries in Secrets Manager.

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 services implemented in this solution. For more information, refer to AWS service quotas.

To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has CloudFormation quotas that you should be aware of when launching the stack for 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 Users Guide*.

Deploy the solution

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

Prerequisites

AMC workflow executions

To execute AMC workflows, you must meet the following prerequisites before launching the stack.

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. Sign in to Amazon Ads to access the AMC console. From the AMC console, record the following parameters for each AMC instance. You will use these throughout the solution. You can find the Instance ID on the account landing page, while the rest are located on the **Instance Info** page.
   - Instance ID
   - Data upload AWS account ID
   - Amazon S3 bucket name
3. For each AMC account, keep note of the following parameters found on the AMC console and Developer API Documentation. You can find the Advertiser ID by selecting your account in the upper-right corner of the AMC console (listed as **ID**), or by looking at the console URL for the value prefixed with ENTITY (example: link:https://advertising.amazon.com/marketing-cloud?entityId=ENTITYX9XX99XX)
   - Advertiser ID
   - Marketplace ID
   - Client ID and Secret ID associated with your Amazon Ads developer account.

Amazon Ads reporting

To retrieve reports for sponsored ads and Amazon DSP campaigns, you must meet the following prerequisites.

- Client ID and Secret ID associated with your Amazon Ads developer account.

- Profile ID associated with an advertising account in a specific marketplace. See the Sagemaker notebook code example to retrieve profile IDs. If you do not have an active one, contact your Amazon Account Team or Amazon Ads API support.

Selling Partner reporting

Before requesting reports using Selling Partner API, follow the Authorization to Selling Partner APIAuthorization to Selling Partner API instructions to meet the following prerequisites.

- Client ID and Client Secret
- Refresh token

Deployment process overview

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

Before you launch the solution, review the cost, architecture, network security, and other considerations discussed earlier in this guide.

> **Note**
>
> If you have previously deployed this solution, see Update the solution for update instructions.

Time to deploy: Approximately 10 minutes

Step 1: Choose your deployment option

- Single account deployment
- Multi account deployment

Step 2: Launch the stack

Step 3: Set AWS Lake Formation permissions

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 AWS 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, see the Anonymized data collection section of this guide.

AWS CloudFormation template

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

View template

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

Note

AWS CloudFormation resources are created from AWS CDK constructs.

IAM policies for installation and admin operation

The solution provides the IAM policies for installing and operating the software. You can use these policies to scope the actions available to a user.

Installation

This solution includes a JSON file named `IAM_POLICY_INSTALL.json`, which is listed in the root folder of the solution source code. You can use this file to create an AWS IAM policy for a user to install the solution.

Operation

IAM policies that can be used to operate the solution as an admin are generated dynamically on stack deployment. Links to the policies can be found in the *Outputs *window of your CloudFormation stack.

> **Note**
>
> The policy generated must be used as a guide. You can review and amend the policy to fit your specific use case.

Step 1: Choose your deployment option

Single-account deployment

We recommend deploying this solution in a single account together with any needed AMC instance Amazon S3 buckets. This ensures that all required dependencies are installed with the solution without additional steps.

Multi-account deployment

If the solution cannot be deployed in the same account as all needed AMC instance Amazon S3 buckets, it can still be deployed in a separate account. However, after deploying the solution, additional resources must be deployed in an AWS account that contains an AMC instance S3 bucket to allow cross-account data lake integration. Refer to Steps to enable cross-account data lake integration.

Step 2: 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 `amazon-marketing-cloud-insights.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 chosen name *must be all lowercase and must be less than 24 characters in length* or the stack will fail to deploy when creating certain resources. For more information about naming character limitations, see IAM and AWS STS quotas in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary.

Parameter	Default	Description
NotificationEmail	*<Requires input>*	Email address to notify subscriber of workflow query results.
ShouldDeployDataLake	*<Requires input>*	Yes - Deploy the data lake. No - Skip data lake deployment.
ShouldDeployMicroservices	*<Requires input>*	Yes - Deploy the Tenant Provisioning Service, Workflow Manager, and Platform Manager Notebooks. No - Skip microservice deployment.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Select the box acknowledging that the template will create IAM resources.
9. Choose **Create stack** 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 3: Set AWS Lake Formation permissions

The following steps must be completed after the stack has successfully deployed. Make sure your stack from the previous step has a CREATE_COMPLETE status before continuing.

Follow these steps to enable AWS Lake Formation to control your AWS Glue Data Catalog resources and to give your IAM role permission to access the tables in the Data Catalog.

1. Sign in to the AWS Lake Formation console.
2. Grant Lake Formation administrative permissions to your IAM role.
   a. In the navigation pane, under **Administration**, choose **Administrative roles and tasks**.
   b. Select **Manage Administrators** and enter your current IAM role.
3. Enable Lake Formation to control your Data Catalog resources.
   a. In the navigation pane, under **Administration**, choose **Data Catalog settings**.
   b. Clear both check boxes and choose **Save**.
4. Give your IAM role permission to access the tables in the Data Catalog.
   a. In the navigation pane, under **Permissions**, choose **Data lake permissions**.
   b. Choose **Grant** in the upper right, and do the following:
      - For **IAM users and roles**, enter your current IAM role.
      - For **LF-Tags or catalog resources**, choose **Named data catalog resources**.
      - For **Databases**, choose your database:
        - *<stack_name>* _datalake_dev_adtech_amc_db.
        - *<stack_name>* _datalake_dev_adtech_ads_report_db.
        - *<stack_name>* _datalake_dev_adtech_sp_report_db.
      - For **Tables**, choose **All Tables**.
      - Keep **Data Filters - Optional** empty.
      - For **Table Permissions**, choose **Super**.
   c. Choose **Grant** at the bottom.
5. Repeat the previous step for any other users who need permission to access the tables in the Data Catalog.

Use the solution

Using the microservices

Use the solution microservices to interact with the Amazon Ads and Selling Partner API, and hydrate your data lake. When you're finished, business stakeholders can use Amazon Athena to access the data returned by the workflow run.

1. Navigate to the AWS CloudFormation console.
2. Select your deployed AMC Insights on AWS stack.
3. Navigate to the **Outputs** tab.
4. Choose the link in the **Value** section, next to the *platformmanagerSageMakerNotebookInstance61A5A1A0 * key.

Authorization

To make requests, you need to first authorize your client credentials, generate a refresh token, and store the token in the AWS Secret Manager. This solution allows you to optionally store multiple OAuth credentials within your secret, and assign a user-defined key referred to as **AuthId** to each set of credentials.

Authorization to Amazon Ads API

This solution creates a notebook to facilitate the Amazon Ads Authorization process.

1. Retrieve Client ID and Secret ID
2. Follow these steps to open the SageMaker notebook instance.
3. Navigate to **amazon_ads** folder in the SageMaker Notebook and open the **Amzon_Ads_Auth.ipynb** file.
4. Follow the instructions in the notebook to complete the authorization process.

Authorization to Selling Partner API

1. Follow the self authorization steps to retrieve a refresh token.

2. Open the Selling Partner secret in AWS Secrets Manager. You can find a link to this secret in your Cloudformation stack Output tab under the key **sellingpartnerSecretsE97D932C**.
3. Store your refresh token, client ID, and client secret in the Secrets Manager. Follow these instructionsto update this value.

   If you only have one set of OAuth credentials, you can update the secret values under Key/value tab.

   If you want to store multiple OAuth credentials, update the secret value under Plaintext tab using a JSON text string. Refer to Managing Multiple Authenticated Credentials for more details.

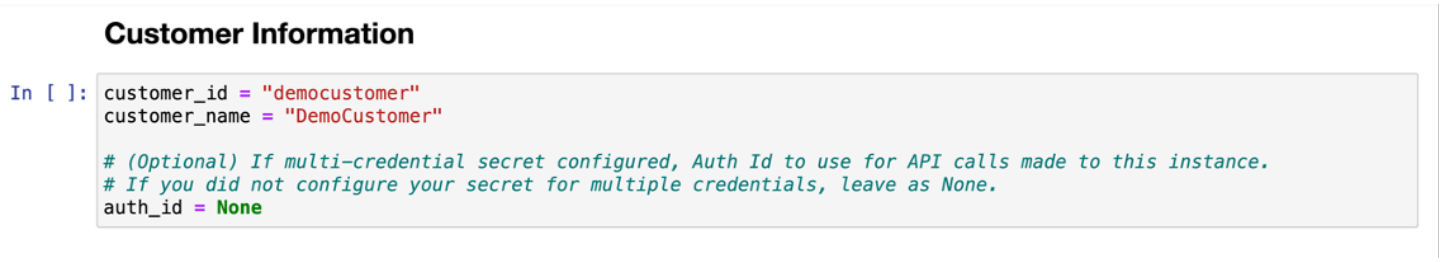
Onboard AMC Instances

AMC provides you with an S3 bucket name you must deploy to receive data back from the sent workflow execution requests. The TPS microservices streamlines this process by deploying the bucket and a corresponding bucket policy, granting only the instance-specific AMC data upload account write access. Therefore, to execute AMC workflows, you must first onboard AMC instances into the solution. The notebook **TPS_Interface.ipynb** provides a step-by-step guide for onboarding one or more AMC instances.

Enter Customer Information

You will need to provide the following customer information in the notebook.

Enter Customer Information



- **customer_id *- A distinct identifier assigned to an AMC instance by you. This can be any unique string that helps identify the instance. When managing AMC workflows, you will input the * customer_id** for which you want to submit request, and the solution will automatically retrieve the AMC instance information associated with that customer.
- **customer_name** - The name of the customer. Like the **customer_id**, this can be a meaningful string that helps identify the customer.

- **auth_id** - An optional parameter that selects right OAuth credentials to use when making API calls to this instance. You only need to specify it when multi-credential secrets configured.

Specify AMC Instance Information

Enter the connection attributes for the AMC instances, see Prerequisites section to locate AMC instance information.

Enter Customer Information

Next, you'll need to choose a deployment pattern for the AMC instance S3 buckets. You have three options:

1. If the bucket doesn't exist in any of your accounts, provide the Region where you want to deploy it and set `bucket_exists` to **false**.
2. If the bucket exists in the current AWS account, provide the region and set `bucket_exists` to **true**.
3. If the bucket exists in a separate AWS account, provide the AWS account ID and set `bucket_account` to the account ID.

Manage AMC Workflows

After authorizing the Amazon Ads API and setting a customer up using TPS microservice, you can open **WFM_Interface.ipynb** and walk through using the Workflow Manager microservice to create, schedule, and run a workflow for that customer. Before running the code examples in the notebook, you will first replace **CUSTOMER_ID** with your customer ID of the onboarded AMC instance.

Workflow Requests

This notebook first provides a collection of code examples that demonstrate how to interact with workflows using AMC's Reporting API. To create a workflow, you'll need to fill in the

`workflow_definition` dictionary with the necessary parameters, including the **WORKFLOW_ID** and workflow definitions.

For a complete list of valid workflow definitions, check out the AMC Reporting API documentation. To update a workflow, you'll need to provide the **WORKFLOW_ID** and updated workflow definitions in the `workflow_definition` dictionary.

If you want to retrieve or delete a workflow, just provide the **WORKFLOW_ID**. After you've executed each cell, a link to the Step Functions execution will be provided, allowing you to view the result of the request on the AWS console.

Workflow Execution Requests

You can use this solution to execute a saved or adhoc workflow. This section of the notebook provides examples for executing an adhoc workflow and canceling a workflow execution. To execute an adhoc workflow, you need to fill in the `adhoc_execution_body` dictionary with the parameters for your adhoc workflow execution, such as the workflow and time window type.

See Amazon Ads for the full list of request body parameters.

If you want to cancel a workflow execution, provide the `workflow_execution_id` variable with the ID of the workflow execution in the code cell.

Workflow Execution Schedules

This solution enables users to schedule workflow executions on a customized cadence. Users will need to provide the following in the notebook:

1. Workflow Execution Body: The request body for a workflow execution. This can be either use an existing workflow execution or an ad hoc workflow. The request body should include the necessary parameters, such as `timeWindowStart` and `timeWindowEnd`, which can be specified using Dynamic Dates functions. See the full list of request body parameters.
2. Schedule Expression: This is a CRON string that specifies the timing of the workflow execution. See Cron expressions.
3. Rule Name: A unique name for the event rule.
4. Rule Description: A brief description of the event rule.

Amazon Ads Reporting

After authorized, you can open the notebook **Ads_Reporting_Interface.ipynb** and follow instructions to request a report from the Amazon Ads Reporting API.

Before generating the report, you'll need to first execute the code cells in the Profiles section of this notebook to retrieve all your connected profiles in the specified regions. Refer to Profilesfor more information.

To create a report request, you need to provide the following values:

1. `profile_id`: The identifier of a profile.
2. `region`: The region where you want to submit the report request. Acceptable values include **North America**, **Europe**, and **APAC**. For Regions, see API endpoints.
3. `table_name`: An optional parameter. If you don't specify a table name, the data lake will use the provided **{Profile Id}-{Report Type Id}** as the destination table.
4. `authId`: An optional parameter that selects right OAuth credentials to use when making API calls to this instance. You only need to specify it when multi-credential secrets configured.
5. `request_body`: Construct the report request body using the information from Creates a report request and Report types documentation. Dynamic Date functions can be used in the `startDate` and `endDate`fields of the request body.

Once you've entered these values, you can execute the code cell to send the report request.

The microservice provides the ability to schedule report requests on a customized cadence. You'll need to define a schedule expression using the Cron format and provide a rule name.

Selling Partner Reporting

After authorized, you can open the notebook **Selling_Partner_Reporting_Interface.ipynb** and follow instructions to request a report from Selling Partner Reporting API.

To create a report request, you need to provide the following values:

1. `authId`: An optional parameter that selects right OAuth credentials to use when making API calls to this instance. You only need to specify it when multi-credential secrets configured.

2. `region`: The region where you want to submit the report request. Acceptable values include **North America**, **Europe**, and **APAC**. For Selling Regions, see SP - API endpoints.
3. `table_prefix`: An optional parameter that allows you to specify a custom table prefix for the report. If you don't provide a table prefix, the data lake will use the default **{Profile Id}-{Report Type Id}** as the destination table prefix.
4. `request_body`: The report request body, which includes the report type, marketplaces, and any other required parameters. See the Tutorial: Request a report and Marketplace IDs for more information. Dynamic Date functions can be used in the `startDate` and `endDate` parameters.

Once you've entered these values, you can execute the code cell to send the report request.

Additionally, this microservice provides the ability to schedule report requests on a customized cadence. You'll need to define a schedule expression using the Cron format and provide a rule name.

Dynamic Dates

This solution provides the following four functions that enable the creation of dynamic dates based on an offset value. By using these functions, you can create scheduled report ranges adaptive, as they are evaluated at request time. This means that when a scheduled report is generated, the functions produce different dates each time, resulting in a more dynamic and up-to-date report.

1. **NOW()**: Returns the current date and time.
2. **TODAY()**: Returns the current date, with the ability to specify a number of days as an offset.
3. **LASTDAYOFOFFSETMONTH()**: Returns the last day of the previous month, with the ability to specify a number of months as an offset.
4. **FIRSTDAYOFOFFSETMONTH()**: Returns the first day of the month, with the ability to specify a number of months as an offset.
5. **FIFTEENTHDAYOFOFFSETMONTH()**: Returns the 15th day of the month, with the ability to specify a number of months as an offset.

For example, **TODAY(-1)** means one day before the current date, while **FIFTEENTHDAYOFOFFSETMONTH(5)** means the 15th day of the month, 5 months from the current month.

Building QuickSight dashboards

This section details how to build an Amazon QuickSight dashboard with AMC data from your data lake. For more information on using QuickSight features, refer to What is Amazon QuickSight?

Create a QuickSight account

To build your first dashboard, you must create a QuickSight account. If you do not have a QuickSight account already, create one by following the steps in Setting up for Amazon QuickSight.

> **Note**
>
> Lake Formation integration with Amazon QuickSight is supported only for Amazon QuickSight Enterprise edition. Ensure that you have an Enterprise account before continuing with the remaining steps.

Authorize QuickSight to access AWS services

For QuickSight to access Athena, Amazon S3, and Lake Formation, a QuickSight administrator must configure the AWS resource permissions. These permissions apply to all QuickSight users. If you're a QuickSight administrator (in which case you will see the **Manage QuickSight** option in your profile menu at the upper right), you can authorize QuickSight access to AWS services using the following two procedures.

Authorize QuickSight to access Athena and Amazon S3

1. In the QuickSight console, select your profile name and choose **Manage QuickSight**.
2. Navigate to **Security & Permissions**.
3. Under **QuickSight access to AWS services**, choose **Manage**.
4. Find **Athena** in the list. Select the box by Athena, then choose **Next**.
5. Under **S3 Bucket**, choose the solution stage bucket to grant QuickSight read access. The name of the bucket can be found in the **Outputs** section of your CloudFormation stack.
6. Choose **Finish**, and save your settings.

Authorize QuickSight to access Lake Formation database and tables

1. Find the Amazon Resource Names (ARNs) of the QuickSight users and groups that need access to Lake Formation data by following the steps in Authorizing connections through AWS Lake Formation.
2. Grant each user or group access by following the steps in Granting database permissions using the Lake Formation console and the names resource method.

Create a dataset in QuickSight

After you've authorized QuickSight to access AWS services, as described in Authorize QuickSight to access AWS services, you can create custom datasets in QuickSight using Athena by following the steps in Creating a datset using Amazon Athena data.

You can then create, publish, and share your custom dashboard.

Adding new datasets

This section details how to extend the solution to add a new dataset and its custom transformation to the deployed insights pipeline.

1. Fork the solution's repository and clone the forked repository.
2. Specify configurations in dictionary format for a new dataset in file `source/infrastructure/datasets_parameters.json`, for example:

```
{
  "dev": [
    {
      "dataset": "newdataset",
      "pipeline": "insights",
      "config": {
        "stage_a_transform": "new_dataset_light_transform",
        "stage_b_transform": "new_dataset_heavy_transform"
      }
    }
  ]
}
```

3. Create custom transformation code for the dataset to be used in stage A and stage B Step Functions for processing the dataset. Place the transformation code under the following paths for stage A and stage B, respectively:

```
source/infrastructure/data_lake/lambda_layers/data_lake_library/python/
datalake_library/transforms/stage_a_transforms/new_dataset_light_transform.py

source/infrastructure/data_lake/lambda_layers/data_lake_library/python/
datalake_library/transforms/stage_a_transforms/new_dataset_heavy_transform.py
```

> **Note**
>
> The `new_dataset_light_transform.py` and `new_dataset_heavy_transform.py` are actual transformation applied in stage A and stage B, and the filename is a reference for the Serverless Data Lake Framework (SDLF) pipeline to pick the transformation for the dataset. The names of the files have to match the configurations specified in `stage_a_transform` and `stage_b_transform` in `datasets_parameters.json` of step 2.

4. Create script for a glue job in Python and place the script in the following path:

```
source/infrastructure/data_lake/glue/lambdas/sdlf_heavy_transform/adtech/
<DATASET_NAME>/main.py
```

> **Note**
>
> The `DATASET_NAME` is a reference to create Glue job for the dataset. The `DATASET_NAME` must match the configurations specified in `dataset` in `datasets_parameters.json` of step 2.

5. Follow the steps in the README.md file to deploy or update AMC Insights on AWS.

Downloading the user scripts

1. Navigate to the CloudFormation console.
2. Select your deployed AMC Insights on AWS stack.
3. Navigate to the **Outputs** tab.
4. Copy the command in the **Value** section, next to the **UserScriptOutput** key.

Note

This command uses the AWS CLI to copy files locally from your deployed solution S3 Artifacts bucket. Ensure you have a compatible version of the AWS CLI installed to run this command.

1. Paste the command into a new terminal session and run it.
2. After running successfully, you will now see a copy of the `amc_insights_user_scripts` folder in your local directory.
3. Refer to the sections Test Cleanup, and Update for instructions on how to use each script.

Monitor the solution with Service Catalog AppRegistry

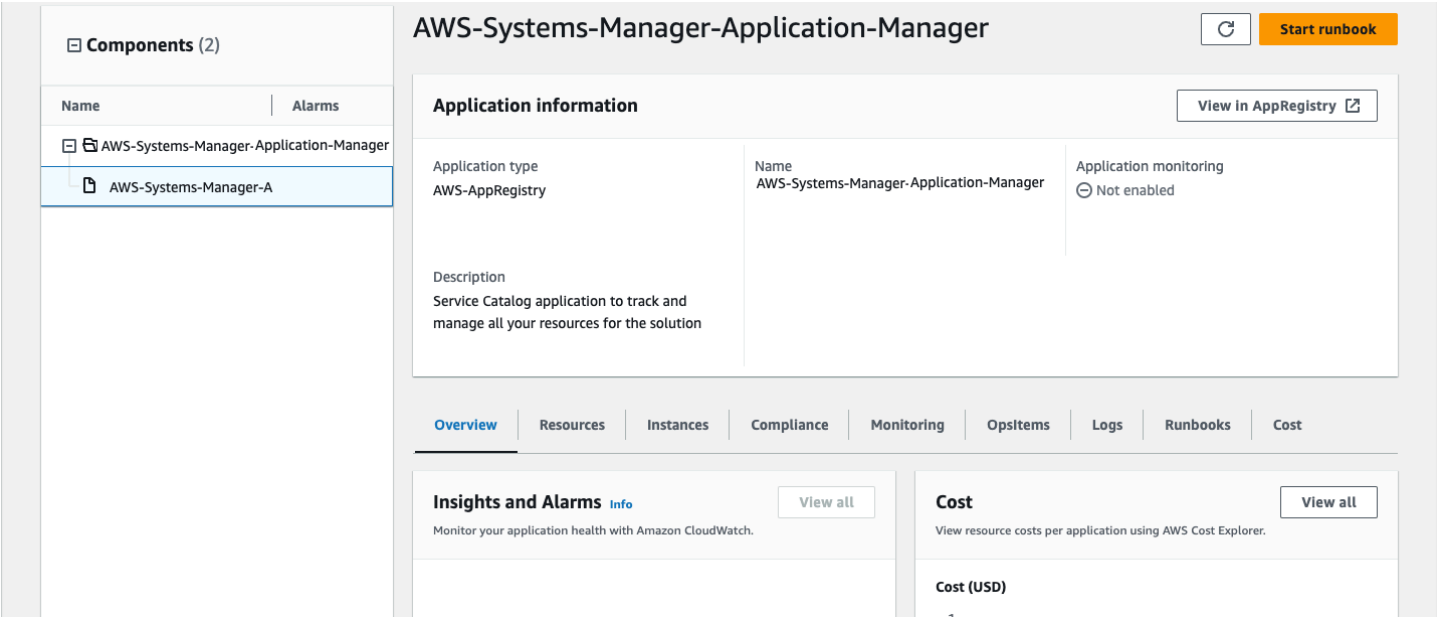
This solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager.

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 (such as deployment status, CloudWatch alarms, resource configurations, and operational issues) in the context of an application.

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

Depicts an AWS Solution stack in Application Manager



Activate CloudWatch Application Insights

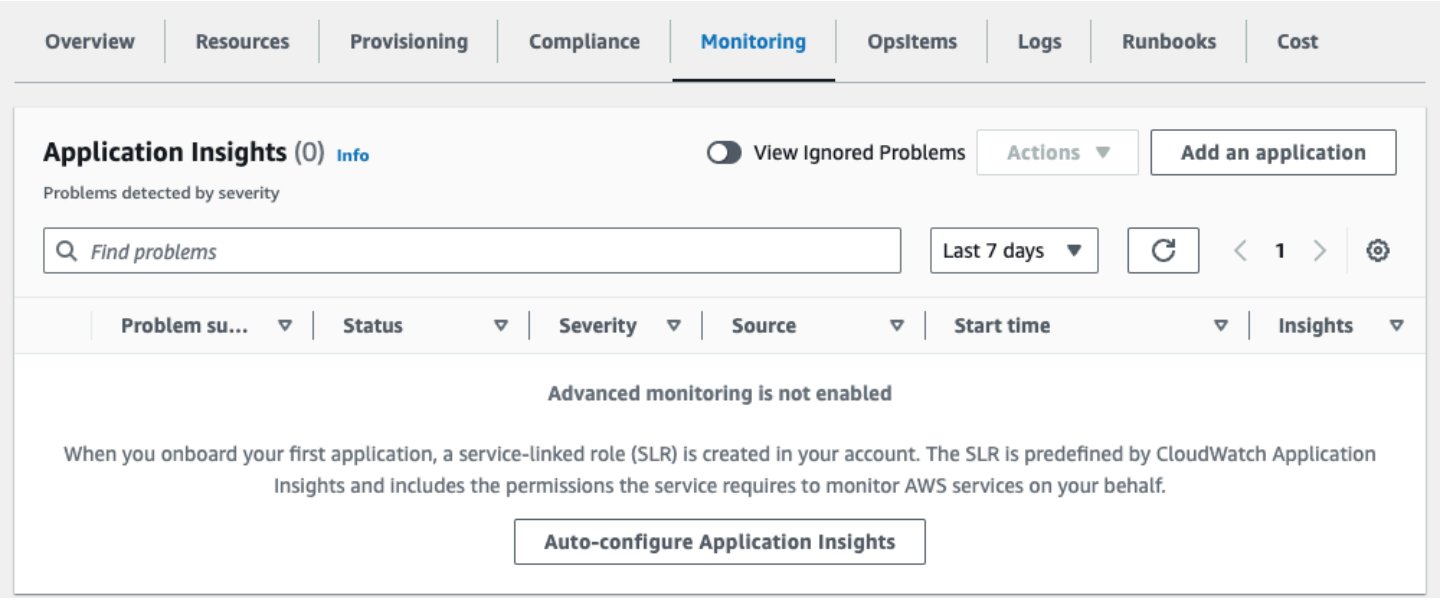
1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.

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.

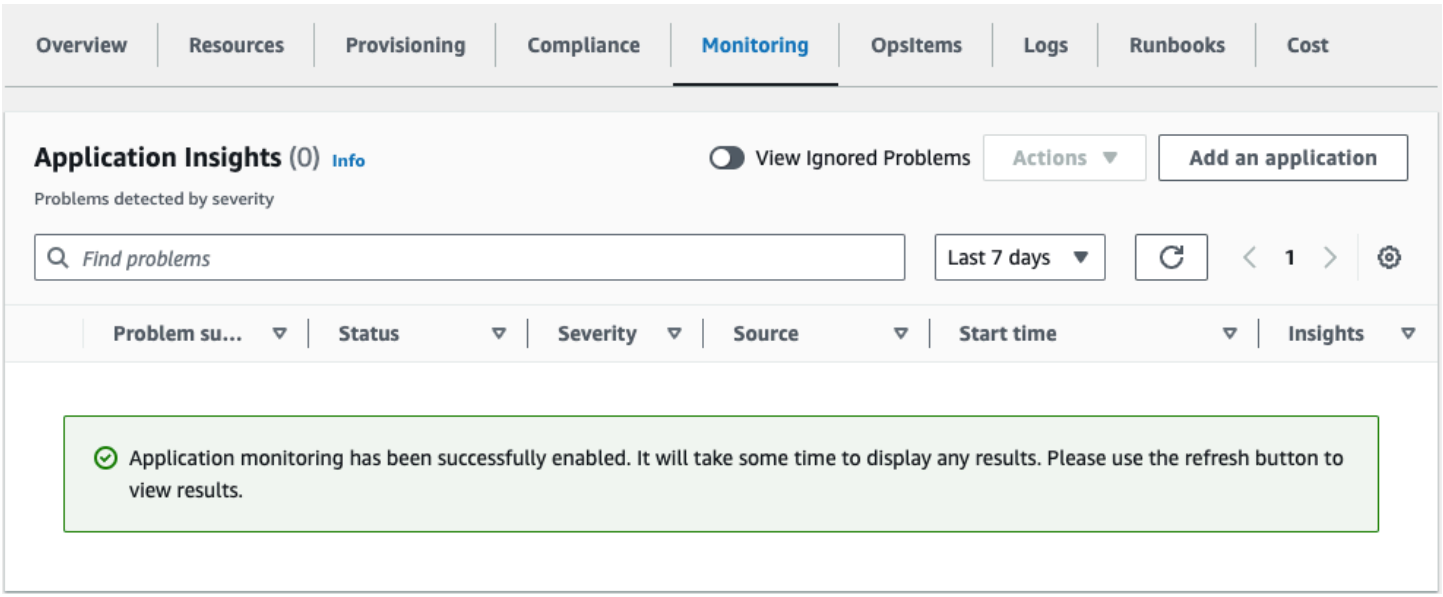
4. In the **Components** tree, choose the application stack you want to activate.
5. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Insights**.

Application Insights dashboard showing no detected problems and advanced monitoring not enabled.



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

Application Insights dashboard showing successful monitoring activation message.

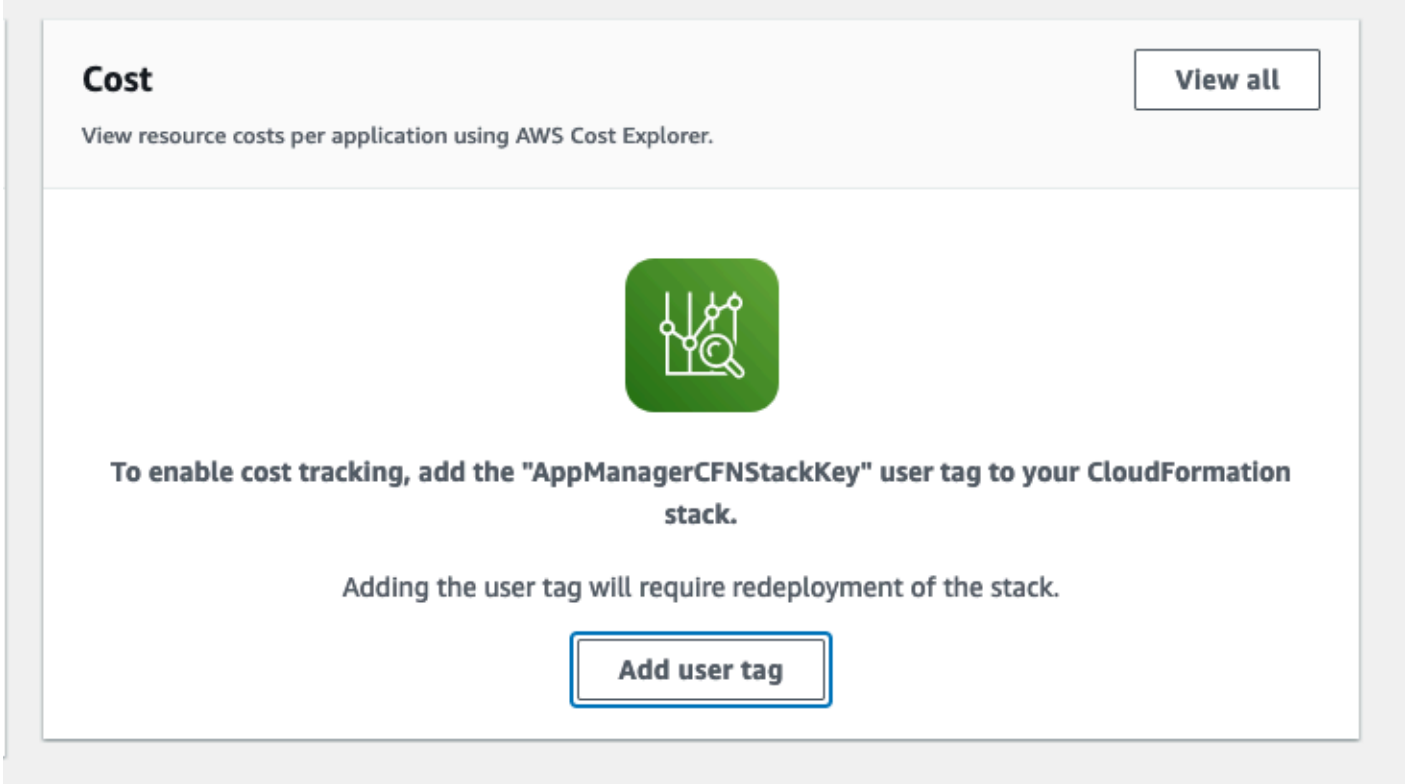


Confirm cost tags associated with the solution

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.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.

Screenshot depicting the Application Cost add user tag screen



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 confirm the cost tags associated with this solution, you must activate the cost allocation tags 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. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time.

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

Update the solution

Updating from version 1.x.x

This update does not support regular stack updates. Instead, deploy a new stack and backfill data from the original stack to the new one. We recommend deploying the new stack in parallel with the original stack, verifying the data and operational consistency of the new one, and deleting the original stack.

Updating from version 2.x.x

1. Follow Updating from other versions to update your main application stack.
2. After updating, follow Using the microservices to authenticate with OAuth for the Amazon Ads API.
3. After authenticating, all previously onboarded instances must be onboarded again through the `TPS_Interface.ipynb` notebook. Unless altering your deployment configuration, use the deployment pattern for existing AMC buckets.

Updating from other versions

1. Log in to AWS CloudFormation console, select your existing `amc-insights` stack, and select **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. Choose **Next**.
5. On the **Configure stack options page**, choose **Next**.
6. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create IAM resources.

7. 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 an UPDATE_COMPLETE status after a few minutes.

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.

Logging

Resources in the stack send logging output to CloudWatch Logs. Entries in these logs can help uncover a misconfiguration or other problem with the stack.

To review log output for different resources:

1. Navigate to the CloudWatch console.
2. In the **Logs** menu, select **Log Groups**. Log group names for the stack begin with the service prefix, followed by the stack name and resource name.

```
/aws/lambda/<StackName>-<LambdaFunctionName>
/aws/vendedlogs/states/<StackName>-<StateMachineExecution>
```

3. Use the search filter at the top of the page to find all the log groups for a stack.

> **Note**
>
> All logs deployed by the solution are configured to never expire. These settings can be changed by editing the retention policy of each log group in the CloudWatch console.

CloudTrail

The stack creates a multi-Region CloudTrail trail that stores data to an Amazon S3 bucket.

In the CloudTrail console, choose **Event history** to find specific API calls relating to metadata including the action, resource name, and type.

Alarms

This stack creates several CloudWatch Alarms to monitor Lambda function invocation and SQS queue processing.

```
<StackName>-<LambdaFunctionName>-lambda-alarm-throttles
```

- The alarm state is ALARM when the named Lambda function encounters a concurrent run limit within a 60-second period.
- The alarm returns to OK if no throttles are encountered for 60 seconds.
- You might be encountering the burst run limit for Lambda.
- You might need to increase the Lambda function concurrent run limit for your account's Region.

```
<StackName>-<LambdaFunctionName>-lambda-alarm-errors
```

- The alarm state is ALARM when the named Lambda encounters a runtime run error within a 60-second period.
- The alarm returns to OK if no errors are encountered for 60 seconds.
- This can be caused by a misconfiguration or failure of resources used by the Lambda Function.
- This can be caused by a bug in the Lambda function's code.

```
<stack-name>-<QueueName>-dlq-a-alarm
<stack-name>-<QueueName>-dlq-b-alarm
```

- The alarm state is ALARM when one or more SQS messages visible in the queue.
- The alarm returns to OK if no messages are visible in the queue.
- Message arriving in this queue can be caused by a misconfiguration or failure of resources used by a related Lambda function. This may be caused by a bug in a Lambda Function's code.

Test

This stack includes a set of functional tests that can be used to help verify successful deployment and troubleshoot issues. You can run these tests by using the following steps.

1. Download the Amazon Marketing Cloud Insights user scripts.
2. Open a new terminal session and navigate to the `test_scripts` directory.
3. Run the following command. Replace the *<stack-name>*, *<profile-name>*, and *<region-name>* variables.

```
$ sh run-test.sh --<stack-name> STACK_NAME --<profile-name> PROFILE --<region-name>
 REGION
```

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 AWS CloudFormation console.
2. On the **Stacks** page, select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether AWS CLI is available in your environment. For installation instructions, see 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>
```

Deleting the Amazon S3 buckets

This solution is configured to retain the solution-created Amazon S3 bucket (for deploying in an opt-in Region) if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete this S3 bucket if you do not need to retain the data. Follow these steps to delete the Amazon S3 bucket.

1. Sign in to the Amazon S3 console.
2. Choose **Buckets** from the left navigation pane.
3. Locate the *<stack-name>* Amazon S3 buckets.
4. Select the Amazon S3 bucket and choose **Empty to remove all data from the bucket**.
5. Select the Amazon S3 bucket and choose **Delete** to permanently remove the bucket from your account.

To delete the Amazon S3 bucket using AWS CLI, run the following command:

```
$ aws s3 rb s3://<bucket-name> --force
```

Deleting the data lake administrators

A data lake administrator is an IAM user; or IAM role that can view all metadata in the AWS Glue Data Catalog and grant permissions on the data resources. After uninstalling the solution, the IAM user; or IAM role created by the solution are deleted, but the data lake administrators in AWS Lake Formation console may not reflect the deletion. You can manually delete the data lake administrators from AWS Lake Formation console by following these steps.

1. Sign in to the AWS Lake Formation console.
2. Choose **Administrative roles and tasks.**
3. Choose **Manage Administrators** in **Data lake administrators.**

Remove the IAM role created by the solution from the data lake administrators list.

Using the cleanup scripts

When you decide to no longer use the solution and not retain the data, delete all the resources deployed by this solution to prevent charges for them. After uninstalling the solution, besides the manual deletion steps described in Uninstall the solution, you can use the cleanup scripts provided in the repo to permanently delete all remaining resources in services like CloudWatch, S3, DynamoDB, KMS, SQS, Lambda, EventBridge, CloudFormation, and Lake Formation. To delete the resources:

1. Download the Amazon Marketing Cloud Insights user scripts.
2. Open a new terminal session and navigate to the `cleanup_scripts` directory.
3. Run the following command. Replace the *<stack-name>*, *<profile-name>*, and *<region-name>* variables.

```
$ sh run-delete-resources.sh --<stack-name> STACK_NAME --<profile-name> PROFILE --
<region-name> REGION
```

Developer guide

This section provides the source code for the solution.

Source code

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

The Amazon Marketing Cloud Insights on AWS templates are generated using the AWS CDK. See the README.md file for additional information.

Supplemental topics

Steps to enable cross-account data lake integration

If the solution was not deployed in the same account as the Amazon **S3 bucket name**, you must complete these additional steps to use the data lake. Complete these steps after onboarding a customer using the Tenant Provisioning Service microservice.

> **Important**
>
> You must repeat these steps for every applicable customer.

1. Navigate to the CloudFormation console.
2. Select the stack name of the target customer to open the **Stack info** window, and choose **Outputs** on the stack menu bar. Then, copy the value for the key: `CrossAccountDataLakeTemplateUrl`.
3. Sign in to the **Connected AWS Account ID** in the Region that contains the target AMC instance S3 bucket, and navigate to the CloudFormation console.
4. Choose the **Create stack dropdown**, and select **With new resources (standard)**.
5. Keep all default selections and paste the `CrossAccountDataLakeTemplateUrl` value copied from step 4 into the **Amazon S3 URL** field.
6. Choose **Next**, enter a stack name, and choose **Next** again.
7. Review and choose **Submit**.

   If your AMC Instance S3 bucket is KMS-encrypted you must grant permission using the following steps (otherwise you may stop here):

8. Navigate to the S3 console and select your AMC Instance S3 bucket.
9. Choose **Properties** on the bucket menu bar, and select **Encryption Key ARN** under the **Default encryption** section.
10. Select **Edit** next to Key policy, and add the following statement to the policy, replacing *[replaceable]* `<AWS_ACCOUNT_ID>` with the solution deployment account, then save the changes.

```
{
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:root"
            },
            "Action": "kms:Decrypt",
            "Resource": "*"
}
```

Retrieving Client ID and Client Secret

1. Set up an Amazon Ads developer account and create a Login with Amazon (LwA) application.
2. Create a new security profile or use an existing profile.
3. From the security profile in use, select **Show Client ID and Client Secret**, or choose the settings icon and download.

See the Create a LwA application documentation for more information.

Managing Multiple Authenticated Credentials

If your API-backed resources span across multiple authenticated accounts, this solution provides the ability to manage your credentials from a single deployment. When adding your credentials through Secrets Manager, you can optionally add multiple pair of credentials by mapping each one to a unique **AuthId** value:

```
{
 “my_auth_id_1” : {
  “client_id”: “”,
  “client_secret”: “”,
  “authorization_code”: “”,
  “refresh_token”: “”,
  “access_token”: “”
},
“my_auth_id_2”: {
  “client_id”: “”,
  “client_secret”: “”,
  “authorization_code”: “”,
  “refresh_token”: “”,
```

```
  “access_token”: “”

}
}
```

You can now pass your **AuthId** in with your requests to ensure the right credentials are used. Refer to the microservice notebooks for examples of how to include this value.

> **Note**
>
> Secrets Manager is limited in the maximum size of a secret value. Refer to quotas to ensure you do not have more credentials than can fit in a single secret.

SageMaker notebook instance lifecycle configuration

This solution implements a lifecycle configuration for the SageMaker notebook instance, which automatically terminates an idle session after one hour to prevent unnecessary instance compute charges.

Steps to modify the auto-stop configuration:

1. Sign in to the Amazon SageMaker console.
2. In the left navigation pane, select **Notebook instances** and select the notebook instance created by this solution.
3. Choose **Actions**, then select **Instance settings**.
4. Under **Instance settings**, select **Lifecycle configuration** and edit the script of the existing configuration.
5. You can modify the value of the `--time` parameter in the command to set the desired idle time in seconds before auto-stop: `(crontab -l 2>/dev/null; echo "*/5 * * * * $(which python) $PWD/autostop.py --time 3600 --ignore-connections >> /var/log/autostop.log 2>&1") | crontab -`

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.

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. When invoked, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each Amazon Marketing Cloud Insights on AWS deployment
- **Timestamp** - Data-collection timestamp
- **Lambda Data** - Count of invocations for each solution lambda function

AWS owns the data gathered through 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 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:

```
"Solution": {
"Data": {
"SendAnonymizedData": "Yes"
    }
}
```

to:

```
"Solution": {
"Data": {
"SendAnonymizedData": "No"
```

```
    }
}
```

4. Sign in to the AWS CloudFormation console.
5. Choose **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 Choose your deployment option in the Deploy the solution section of this guide.

Contributors

- Ian Downard
- Chris Geiger
- Immanuel George
- Andrew Marriott
- Alessandro Narciso
- Yang Qin
- Jim Thario

Revisions

Publication date: *December 2022*.

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.