Implementation Guide

Maintaining Personalized Experiences with Machine Learning

Maintaining Personalized Experiences with Machine Learning: Implementation Guide

Table of Contents

Deploy a mechanism to keep personalized recommendations up-to-date

Publication date: September 2021. Check the CHANGELOG.md file in the GitHub repository to see all notable changes and updates to the software. The changelog provides a clear record of improvements and fixes for each version.

This implementation guide provides an overview of the Maintaining Personalized Experiences with Machine Learning solution, its reference architecture and components, considerations for deployment and configuration steps for deploying this solution to the Amazon Web Services (AWS) Cloud.

Maintaining Personalized Experiences with Machine Learning solution streamlines and accelerates development by providing automated pipeline construction, automated personalization model including: configuration, training, retraining, and deployment, as well as improved visibility into model performance, and advanced error handling mechanisms. It helps you build personalized experiences with Amazon Personalize for your product portfolio and provide real-time, curated experiences across digital channels.

Implementing Maintaining Personalized Experiences with Machine Learning solution aims to increase your user engagement metrics, click-throughs, and conversion rates by providing up-to-date product recommendations, personalized product re-rankings, user segmentation, and customized direct marketing.

Personalization opportunities exist in multiple areas along the customer journey including:

- **Discoverability** - Helping consumers easily and quickly discover products and content
- **Acquisition and retention** - Attracting and retaining consumers in a digital environment
- **Engagement** - Understanding, measuring, and increasing time spent engaging with products and content
- **Efficiencies and revenue** - Increasing average revenue per user

This implementation guide provides an overview of the Maintaining Personalized Experiences with Machine Learning solution, its reference architecture and components, considerations for deployment and configuration steps for deploying this solution to the Amazon Web Services (AWS) Cloud.

The guide is intended for IT architects, developers, DevOps, and data analysts who have practical experience architecting in the AWS Cloud and want to implement Maintaining Personalized Experiences with Machine Learning in their environment.

> **Note**
>
> This solution is not recommended for handling regulated data such as PII, HIPAA, and GPDR when deployed in production.

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

If you want to . . .	Read . . .
Know the cost for running this solution.	Cost
Understand the security considerations for this solution. Security responsibilities are shared between you and AWS.	Security
Know how to plan for quotas for this solution. AWS CloudFormation, AWS Lambda and AWS Step Function quotas may apply.	Quotas
Know which AWS Regions are supported for 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 Maintaining Personalized Experiences with Machine Learning solution accelerates the development and deployment of personalization workloads by leveraging the functionalities of the Amazon Personalize service and streamlining production and maintenance of Amazon Personalize solutions.

It provides end-to-end automation for Amazon Personalize through the entire lifecycle of a workload, and its features include:

- Automating the creation of Amazon Personalize recommenders, solutions, and solution versions to baseline model performance (by comparing user-configured model offline metrics over time).
- Presenting the results in an Amazon CloudWatch dashboard.
- Automating the scheduled retraining and update of Amazon Personalize solution versions to assess their performance over time, as typical personalization workloads benefit from full model retraining every one to five days.
- Automating the scheduled creation of batch recommendations and batch user segmentation.
- Integration with AWS Service Catalog AppRegistry and Application Manager, a capability of AWS Systems 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 Application Manager. With this integration, you can centrally manage the solution's resources and enable application search, reporting, and management actions.

Use cases

You can use Amazon Personalize to:

Drive better user engagement

Use personalized user recommendations to help drive better engagement and conversion (USER_PERSONALIZATION recipes).

Improve customer search experience

Personalized curated lists or search results for your users improve the customer experience and increase customer loyalty and engagement. Amazon Personalize helps create a personalized

list by re-ranking a collection of input items based on predicted interest level for a given user (PERSONALIZED_RANKING recipes).

Keep users engaged with trending or popular items

If your customers highly value what other users are interacting with, Amazon Personalize can help recommend trending or popular items. Common uses include recommending viral social media content, breaking news articles, or recent sports videos (POPULAR_ITEMS recipes).

Increase user conversion by recommending similar items

Recommending similar items can help your customers discover items and can increase user conversion rate. Amazon Personalize can help recommend similar items, such as items frequently bought together or movies that other users have also watched (RELATED_ITEMS recipes).

Targeted marketing campaigns through user segments

Getting user segments can help you create advanced marketing campaigns that promote different items to different user segments based on the likelihood that they will take an action. Amazon Personalize can help generate segments of users based on item input data, such as users who will most likely interact with items with a certain attribute (USER_SEGMENTATION recipes).

To increase the adoption for Amazon Personalize, you can use the Maintaining Personalized Experiences with Machine Learning solution to:

Create automated Amazon Personalize recommendation pipelines

Select an appropriate recipe per your use case and create automated pipelines as simply as providing json files and data in a S3 bucket.

Run retraining and inferencing via scheduled cron jobs

As a part of the json configuration, schedules can be setup to retrain the model fully/update it at a desired frequency. Campaigns, batch inferencing, etc. can also be made to use the updated models.

Advanced error handling and status updates

The solution provides error messages and job completion statuses via an SNS topic and email notification. This simplifies error handling and status updates and you can receive completed statuses for scheduled jobs, and in case of failures, an AWS XRay link to debug the issue.

Concepts and definitions

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

recipe

recipes are Amazon Personalize algorithms that are prepared for specific use cases, such as USER_PERSONALIZATION or POPULAR_ITEMS recipes.

solution

a solution refers to the combination of an Amazon Personalize recipe, customized parameters, and one or more solution versions (trained models). After you create a solution with a solution version, you can create a campaign to deploy the solution version and get recommendations.

solution version

a solution version is trained machine learning model that can be deployed to get customer recommendations. It's the realized form of a solution definition.

campaign

campaigns are deployed solution versions.

batch inference job

a batch inference job gets batch item recommendations for users based on input data from Amazon S3.

batch segment job

if you used a USER_SEGMENTATION recipe, you can create batch segment jobs to get user segments with your solution version.

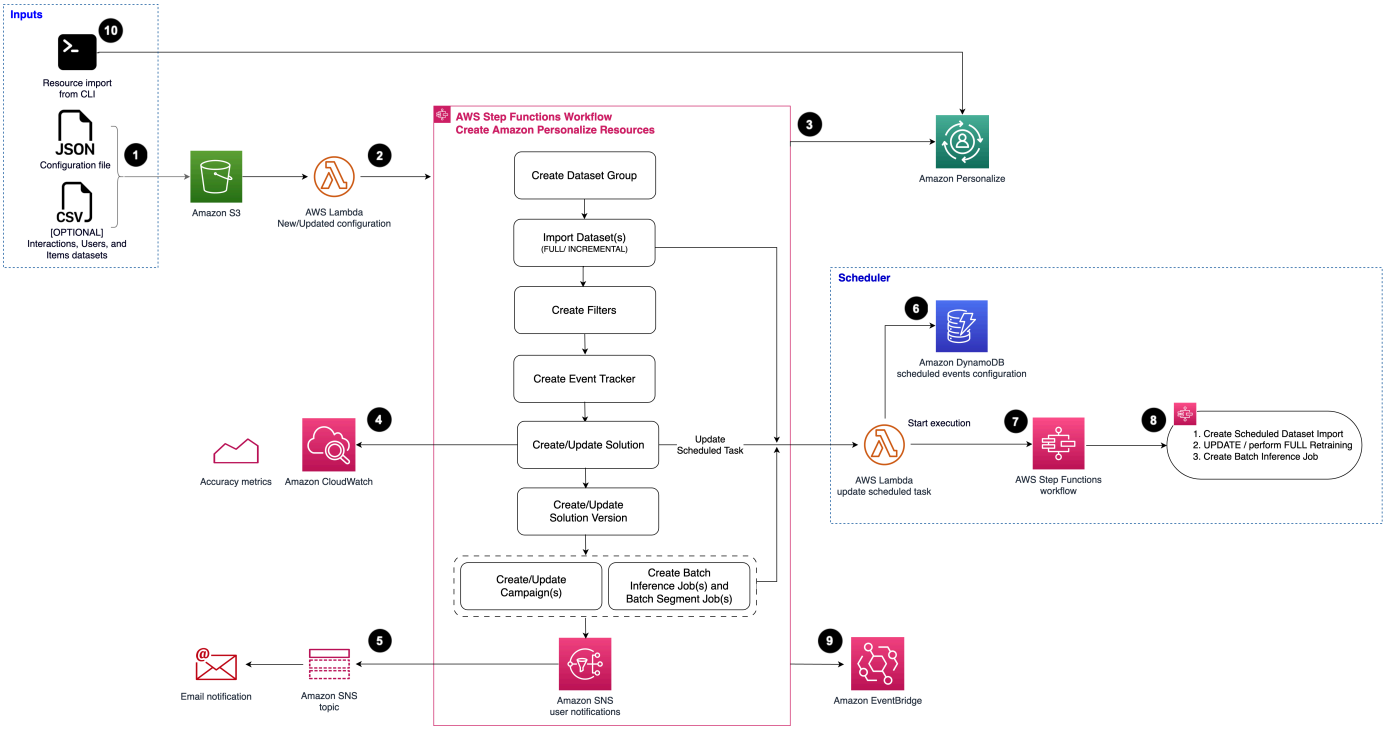
> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary and for more information on Amazon Personalize, see the Developer Guide.

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:



Maintaining Personalized Experiences with Machine Learning architecture

The AWS CloudFormation template deploys the following infrastructure:

1. An Amazon S3 bucket used to store personalization data and configuration files.
2. An AWS Lambda function initiated when new or updated personalization configuration is uploaded to the personalization data bucket.
3. An AWS Step Functions workflow to manage all of the resources of an Amazon Personalize dataset group (including datasets, schemas, event tracker, filters, recommenders, solutions, campaigns, batch inference jobs, and batch segment jobs).

4. Amazon CloudWatch metrics for Amazon Personalize for each new trained solution version are added to help you evaluate the performance of a model over time.
5. An Amazon Simple Notification Service (SNS) topic and subscription to notify an administrator when the maintenance workflow has completed via email.
6. Amazon DynamoDB tracks the scheduled events configured for Amazon Personalize to fully or partially retrain Amazon Personalize solutions, import or reimport datasets, and perform batch inference jobs.
7. An AWS Step Functions workflow tracks the current running scheduled events, and invoke step functions to perform Amazon Personalize solution maintenance (creating new solution versions, updating campaigns), import updated datasets, and perform batch inference.
8. A set of maintenance AWS Step Functions workflows are provided to:
   - Create new dataset import jobs on schedule.
   - Perform Amazon Personalize solution FULL retraining on schedule (and update associated campaigns).
   - Perform Amazon Personalize solution UPDATE retraining on schedule (and update associated campaigns).
   - Create batch inference jobs.
9. An Amazon EventBridge event bus, where resource status notification updates are posted throughout the AWS Step Functions workflow.
10. A command line interface (CLI) allows you to import and establish schedules for resources that already exist in Amazon Personalize.

> **Note**
>
> AWS CloudFormation resources are created from AWS CDK constructs.

AWS Well-Architected design framework

We designed this solution 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 we applied the design principles and best practices of the Well-Architected Framework when building this solution.

Operational Excellence

1. **Use of Solution Constructs** - Beyond being a fully serverless automation workflow to ensure no single point of failure, the Maintaining Personalized Experiences with Machine Learning solution uses the AWS Solutions Constructs (a library of pre-built multi-service, well-architected patterns for quickly defining solutions in code to create predictable and repeatable infrastructure) where applicable.
2. **Monitoring systems** - CloudWatch Metrics are published for each Dataset Group and Solution Version combination to track offline metrics over time. All metrics published by Amazon Personalize are published by the solution as CloudWatch Metrics, and can (optionally) be baselined against the Popularity-Count offline metrics.

   These metrics can be visualized in a CloudWatch dashboard (which can be enabled on user request by configuration change).
3. **Continual improvement** - Customers can effectively test out changes, troubleshoot problems and test new hypotheses when creating new solution versions.
4. **Automating changes** - The solution can be updated to activate new functionality by updating the CloudFormation template in place.

   AWS Solutions Constructs patterns and the AWS CDK are used to automate the creation of the CloudFormation templates required by the solution.
5. **Responding to events** - The solution provides updates to the operator via an email through subscription to an SNS topic. This allows an operator to take action as issues are discovered in solution version and campaign creation (for example, insufficient data, misconfiguration), and be notified when specific conditions are met (e.g. solution version + campaign are ready).
6. **Standards for daily ops** - By standardizing how Amazon Personalize resources are created, and how bulk data is ingested into the service, operational procedures can be built around the Maintaining Personalized Experiences with Machine Learning solution. In the future, this will enable operators to integrate with other business systems (e.g. Segment as a CDP, Optimizely for A/B testing).

Security

1. **Confidentiality of data** - The data is encrypted at rest (using SSE-S3) in the customer bucket and it is encrypted at rest (using SSE-S3) in the Amazon Personalize server-side. Users can optionally enable usage of one of their own AWS KMS keys (the solution supports this).

2. **Integrity of data** - This solution leverages the AWS global infrastructure, which is built around AWS Regions and Availability Zones.

   AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected by low-latency, high-throughput, and highly redundant networking.

   Apart from this, Amazon S3 which is used in the solution, is designed for 11 9s of durability.
3. **Access management** - Access is restricted to users through the use of IAM. Amazon Personalize uses an IAM role and policy to load data from S3. The use of of AWS managed policies is avoided where possible to provide the principle of least priviledge

Reliability

1. **Common failures** - Protection against task failures is provided by detection of common failures and sending alerts to an SNS topic, which provides a user notification via email.
2. **AWS state machines** - The Maintaining Personalized Experiences with Machine Learning solution workflow utilizes AWS Step functions. Step functions states can fail for a variety of reasons, including state machine definition errors, task failures, and transient issues. Protection against transient issues is provided by ensuring individual tasks in the step function are idempotent where possible. This allows effective retrial of tasks.
3. **Exponential backoff to prevent throttling** - Exponential backoff is utilized on the Personalize service calls to reduce the likelihood of throttling due to service-side rate limiting. These include:
   - Rate limiting for > 5 pending/ in progress dataset import jobs
   - Rate limiting for > 5 pending/ in progress batch inference jobs
   - Rate limiting for > 5 pending/ in progress solutions versions

Furthermore, the AWS (boto3), the Python SDK used to develop the solution implements client-side API throttling.

Performance Efficiency

1. **Serverless operations** - The Maintaining Personalized Experiences with Machine Learning solution uses a serverless architecture to eliminate our dependency on long-running EC2 instances. This significantly reduces cost of running workflows within AWS.
2. **Default usage** - Where possible, expensive operations are disabled by default (for example, API Gateway caching is not enabled by default).

Cost Optimization

The use of a serverless architecture in the Maintaining Personalized Experiences with Machine Learning solution eliminates the dependency on long-running EC2 instances. This significantly reduces the cost of running workflows within AWS. Combining the solution with the Amazon Personalize Monitor allows campaign TPS to be monitored efficiently and adjusted as required.

Sustainability

Sustainability in the cloud is a continuous effort focused primarily on energy reduction and efficiency across all components of a workload by achieving the maximum benefit from the resources provisioned and minimizing the total resources required.

Maintaining Personalized Experiences with Machine Learning solution uses serverless services (such as, AWS Lambda and AWS DynamoDB) which are aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers, or long-running EC2 instances. This eliminates idle resources, processing, and storage to reduce the total energy required to power customer workloads.

Architecture details

Personalization AWS Step Function workflows

There are three AWS Step Functions personalization workflows included with the AWS solution:

- **<stack_name>-personalize-workflow**

This is the main workflow, invoked when a valid new configuration file is uploaded to the Amazon S3 bucket for personalization data. This workflow creates all declared resources in the configuration file and any schedules associated with the maintenance of those resources.

- **<stack_name>-periodic-solution-maintenance**

This is a workflow that can be initiated on a schedule to perform Amazon Personalize solution maintenance. This maintenance can include (a) Amazon Personalize solution version creation and campaign update and/or (b) batch inference job creation. Amazon Personalize solution versions can be created as new versions (a FULL model retraining) or as updates (UPDATE model).

> **Note**
>
> The UPDATE retraining option in Amazon Personalize can only be used when there is an active solution version created from an input solution using the FULL option and the input solution was trained with the `User-Personalization` or `HRNN-Coldstart` recipe.The first training happens in "FULL" training modes and the subsequent training modes become update when trainingMode is specified. Refer to README.md for more details on the usage.

- **<stack_name>-periodic-dataset-import**

This is a nested workflow that can be initiated to import data on a schedule. Establishing schedules for dataset import is useful when creating batch inferences against data not being collected by an event tracker.

> **Warning**
>
> The dataset import job replaces any existing data in the dataset that was previously imported in bulk.

Scheduler

Maintaining Personalized Recommendations with Machine Learning deploys an AWS Step Function that acts as a scheduler in order to schedule periodic maintenance workflows. Each scheduled item consists of a name, cron-style schedule, step function to invoke, and step function input, and is managed by the main workflow. Each scheduled task configuration is versioned and stored in Amazon DynamoDB. For more information about cron-style schedules, refer to Schedule Expressions for Rules in the *Amazon CloudWatch Events User Guide.*

This solution provides scheduling support for:

- Dataset import (for domain and custom dataset groups)
- Amazon Personalize solution maintenance (for custom dataset groups only)
- Batch inference job creation (for custom dataset groups only)

This solution provides a command line interface (CLI) that can be used to establish schedules for dataset import and solution maintenance for resources in Amazon Personalize that were not created by the solution.

AWS services in this solution

The following AWS services are included in this solution:

AWS service	Description
Amazon S3	**Core.** Used to upload user configuration files (.json), datasets for user, items and interacti ons datasets (.csv).

AWS service	Description
AWS Lambda	**Core.**Used to trigger step function workflows , and run each step in the step function personalize workflow.
Amazon Personalize Service	**Core.**This solution is created around Amazon Personalize service to create dataset groups, datasets, train models using recipes and create solutions and solution versions, campaigns, batch inference and segment jobs, recommenders, etc.
AWS Step Functions	**Core.** Manages all resources of an Amazon Personalize through step function states.
Amazon Simple Notification Service	**Supporting.** Amazon SNS topic and subscript ion help notify an administrator when the maintenance workflow has completed via email, if an error occurs due to invalid configuration upload, etc.
Amazon CloudWatch	**Optional.** Amazon Cloudwatch metrics for Amazon Personalize are added for each new trained solution version to help you evaluate the performance of a model over time.
AWS Systems Manager	**Supporting.** Provides application-level resource monitoring and visualization of resource operations and cost data.

Plan your deployment

This section describes the cost, security, Regions, quotas, and other considerations prior to deploying the solution.

Cost

You are responsible for the cost of the AWS services used while running this AWS solution. As of this revision, the cost for running this AWS solution with the default options for powering content discovery and recommendation through real-time profiling of user preferences and consumption behavior against a 200GB dataset, training daily, with each training taking 20 minutes to complete and consuming 10 training hours per training, while offering real-time inference (**at 10TPS for 24 hours per day**) and exporting a single batch inference for one million users at the end of the month in the US East (N. Virginia) is **$1,603.40**.

This solution's cost is highly dependent on Amazon Personalize transactions per second (TPS). For more information, refer to Amazon Personalize Pricing.

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. This cost estimate does not account for Amazon S3 PUT and GET requests, which can vary depending on how frequently data is accessed in S3.

AWS service	Dimensions	Cost [USD]
Amazon Personalize Training hours	300 training hours x $0.24 USD	$72.00
Amazon Personalize Data Storage	200 GB x $0.05 USD	$10.00

AWS service	Dimensions	Cost [USD]
Amazon Personalize Real-time Inferencing	Real-Time Inference for 24 hrs x 30 days x 10TPS = 7200 TPS hrs 720 TPS-hours x $0.20 USD	$1,440.00
Amazon Personalize Batch Inferencing	1,000,000 batch inferenci ng recommendations x $0.000067 USD	$67.00
Amazon S3	200 GB S3 Standard storage x $0.023 USD	$4.60
AWS Step Functions	100 state transitions for 100 workflow requests =10,000 state transitions per month 4000 Free Tier state transitio ns are available per month, so 6,000 billable state transitio ns 6,000 state transitions x $0.000025 USD	$1.50
Amazon DynamoDB	2 items, 10KB storage, with Point in Time Recovery (PITR) enabled 21 metrics x $0.30 USD 1 GB ingested data for logs x $0.50 USD	$0.20 Metrics: $6.30 Log ingestion: $0.50
Amazon CloudWatch	13 metrics x $0.10 USD for the dashboard	Dashboard: $1.30

AWS service	Dimensions	Cost [USD]
		Total monthly cost: $1,603.40

Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this AWS solution.

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 and the Amazon Personalize Developer Guide.

AWS recommends rotating your KMS keys on a schedule to at least once in 90 days/as per your enterprise's policies. Also, we recommend enabling CloudTrail to monitor and record account activity across your AWS infrastructure, giving you control over storage, analysis, and remediation actions. This can allow you to track a malicious activity in a security incident.

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 AWS solution creates IAM roles that grant the solution's AWS Lambda functions access to create Regional resources.

Supported AWS Regions

Maintaining Personalized Experiences with Machine Learning solution is supported in the following AWS Regions:

Region name
US East (Ohio)
Asia Pacific (Seoul)

Region name
US East (N. Virginia)
Asia Pacific (Sydney)
US West (Oregon)
China (Beijing)
Asia Pacific (Mumbai)
Canada (Central)
Asia Pacific (Singapore)
Europe (Ireland)
Asia Pacific (Tokyo)
Europe (Frankfort)

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, see 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 page in the PDF instead.

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, see AWS CloudFormation quotas in the in the *AWS CloudFormation User's Guide*

AWS Lambda quotas

When managing large numbers of scheduled tasks using this AWS solution, it is possible that the account quota for concurrent tasks might be met. When scaling to hundreds of schedules, consider increasing the quota for concurrent tasks in AWS Lambda. For more information refer to Lambda Quotas in the *AWS Lambda Developer Guide.*

AWS Step Functions quotas

When managing large numbers of scheduled tasks using this AWS solution, it possible that the account quotas for state throttling might be met. When scaling to hundreds of schedules, consider increasing the quota for bucket size and refill rate per second. For more information on these quotas, refer to Quotas in the *AWS Step Functions Developer Guide.*

Resource naming

In order to produce human-readable names in Amazon Personalize and in the AWS solution's Amazon S3 bucket, Maintaining Personalized Recommendations with Machine Learning follows a naming convention for batch inference jobs.

Resource	Naming convention	Maximum resource name length
Batch Inference Job	`batch_[replaceable] <solution_name>\_ %Y_%m_%d_%H_%M_%S`	`solution_name` must not exceed 37 characters in length
Batch Segment Job	`batch_[replaceable] <solution_name>\_ %Y_%m_%d_%H_%M_%S`	`solution_name` must not exceed 37 characters in length

If these limits are exceeded, the AWS solution will notify the subscribed email address of the error in configuration.

Deploy the solution

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

> **Note**
>
> The solution uses SSE-S3 encryption for Amazon S3 buckets and AWS Managed encryption for other services as defaults. If there are specific regulatory or compliance requirements for your use case, we recommend that you review them and make necessary amends in the solution before deploying the solution and uploading any data into the buckets. You can make use of the compliance information provided for the Amazon Personalize service as well. You can make use of the Amazon Personalize service towards this task.

Deployment process overview

Before you launch the solution, review the cost, Architecture overview, network 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 AWS account.

Time to deploy: Approximately five minutes

Use the following steps to deploy Maintaining Personalized Recommendations with Machine Learning on AWS. For detailed instructions, follow the links for each step.

Step 1. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Review the templates parameters and enter or adjust the default values as needed.

Step 2. Use the solution to create and/or maintain resources in Amazon Personalize

- Create a configuration .json file for the Amazon Personalize resources to maintain and upload it to the Amazon S3 bucket deployed by the AWS solution.

Important

This AWS solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this AWS solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Policy.

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 template and deploy the AWS solution. For more information, refer to the Anonymized data collection section of this guide.

AWS CloudFormation template

To automate deployment, this AWS solution uses the following AWS CloudFormation template, which you can download before deployment:

View template button

maintaining-personalized-experiences-with-machine-learning.template - Use this template to launch the solution and all associated components. The default configuration deploys the services listed in AWS services in this solution, but you can customize the template to meet your specific needs.

Note

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

This AWS CloudFormation template deploys Maintaining Personalized Experiences with Machine Learning in the AWS Cloud. You must meet the following prerequisites before launching the stack:

Note

If you have previously deployed this solution, see Update the stack for update instructions.

Step 1. Launch the stack

This automated AWS CloudFormation template deploys the Maintaining Personalized Experiences with Machine Learning solution in the AWS Cloud.

> **Note**
>
> You are responsible for the cost of the AWS services used while running this AWS solution. For more details, visit the Cost section in this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and select the button to launch the `maintaining-personalized-experiences-with-machine-learning.template` AWS CloudFormation template.

   Solution launch button

2. The template launches in the US East (N. Virginia) Region by default. To launch the AWS solution in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > This AWS solution uses the Amazon Personalize service, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Amazon Personalize is available. For the most current availability by Region, refer to the AWS Regional Services List.

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 AWS solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this AWS solution template and modify them as necessary. This AWS solution uses the following default values.

Parameter	Default	Description
Email	*<Optional input>*	The email that receives status notifications from the AWS Step Functions state machines deployed by this solution. If this parameter is blank, you are not notified with your personalization maintenance job results. You must accept the SNS subscription (through the email link that is sent after stack deployment) to activate notifications.

Parameter	Default	Description
Personalize KMS Key ARN	*<Optional input>*	While Amazon Personalize will encrypt your data by default, you can monitor and restrict access to your data by specifying a KMS key in your account that can be used by the Amazon Personalize service. Specifying an AWS KMS key ARN in this parameter allows Amazon Personalize to use that key to protect your data. Revoking, turning off, or modifying the key policy associated with this key may render your data unusable by the Amazon Personalize service. To use your own key, specify the full key ARN, for example: `arn:aws:kms:<region>:<account_id>:key/f8fed2cd-14ab-4ac4-a8a3-57975cbff81b` . The key policy must grant the personalize service Encrypt, Decrypt, GenerateDataKey, and DescribeKey permissions on the key, as well as turn on user permissions for the key. An

Parameter	Default	Description
		example key policy can be found in KMS customer-managed key policy. Leave this parameter blank to have Amazon Personalize use its default encryption configuration for your data.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. 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.
9. 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 five minutes.

> **Note**
>
> In addition to the primary AWS Lambda functions used to create and maintain Amazon Personalize resources, this AWS solution includes additional Lambda functions, that run only during initial configuration or when resources are updated or deleted.
> When you run this AWS solution, you will notice Lambda functions with the stack name you used as a prefix in the AWS console. You must not delete any of these functions, as they are necessary to manage associated resources.

Step 2. Create and maintain Amazon Personalize resources

Use the procedures in User guide to create and maintain resources in Amazon Personalize. The solution configures resources in Amazon Personalize based on configuration .json files uploaded to the S3 bucket deployed by the solution. Resources are configured and maintained based on the contents of this file.

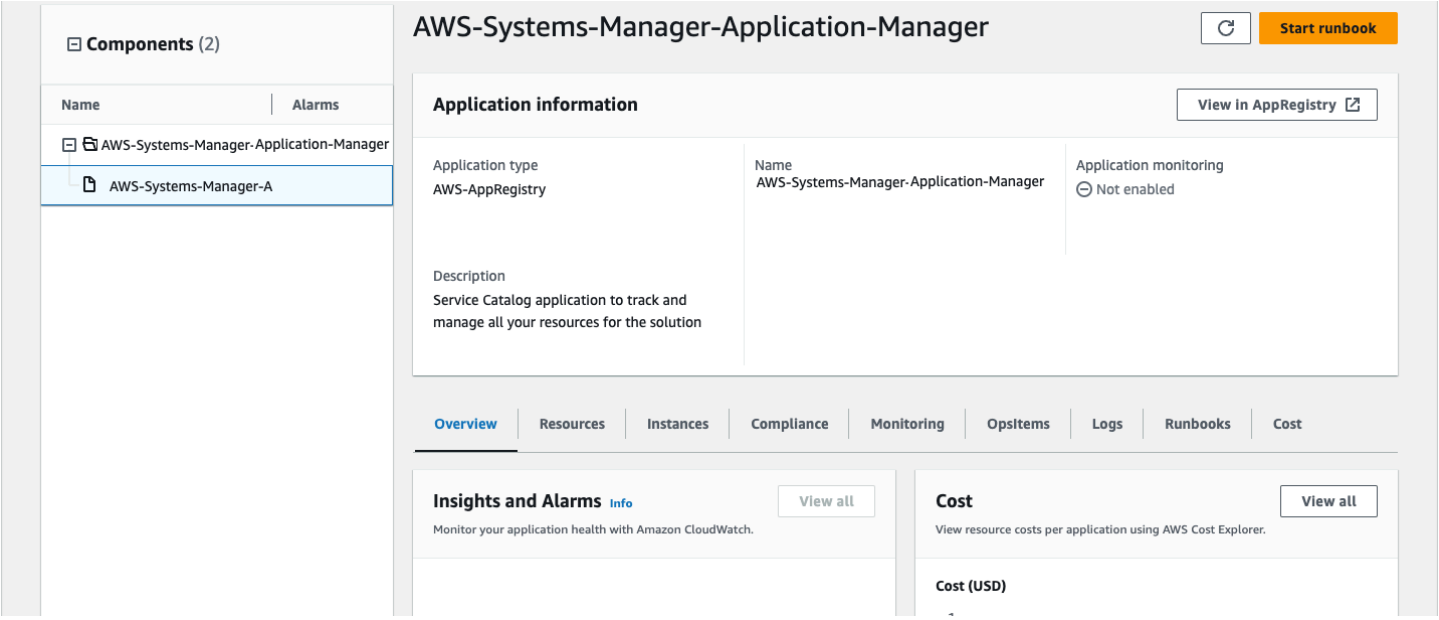
Monitor the solution with Service Catalog AppRegistry

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.



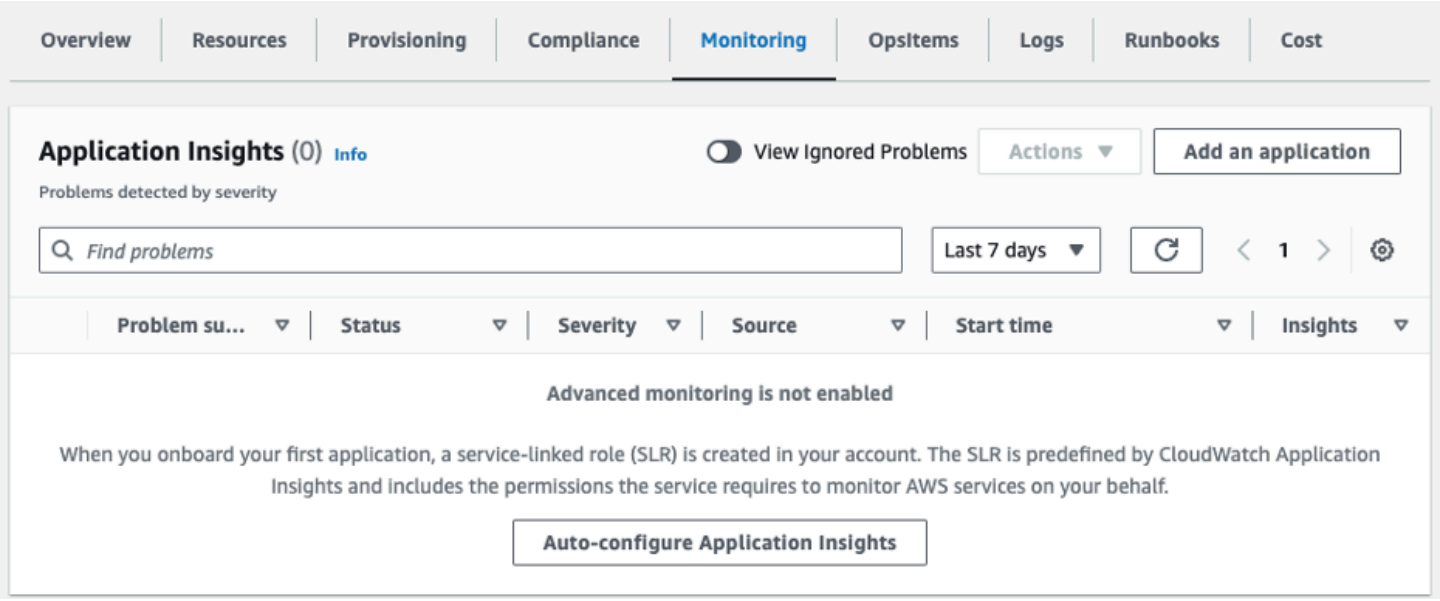
Activate CloudWatch Application Insights

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

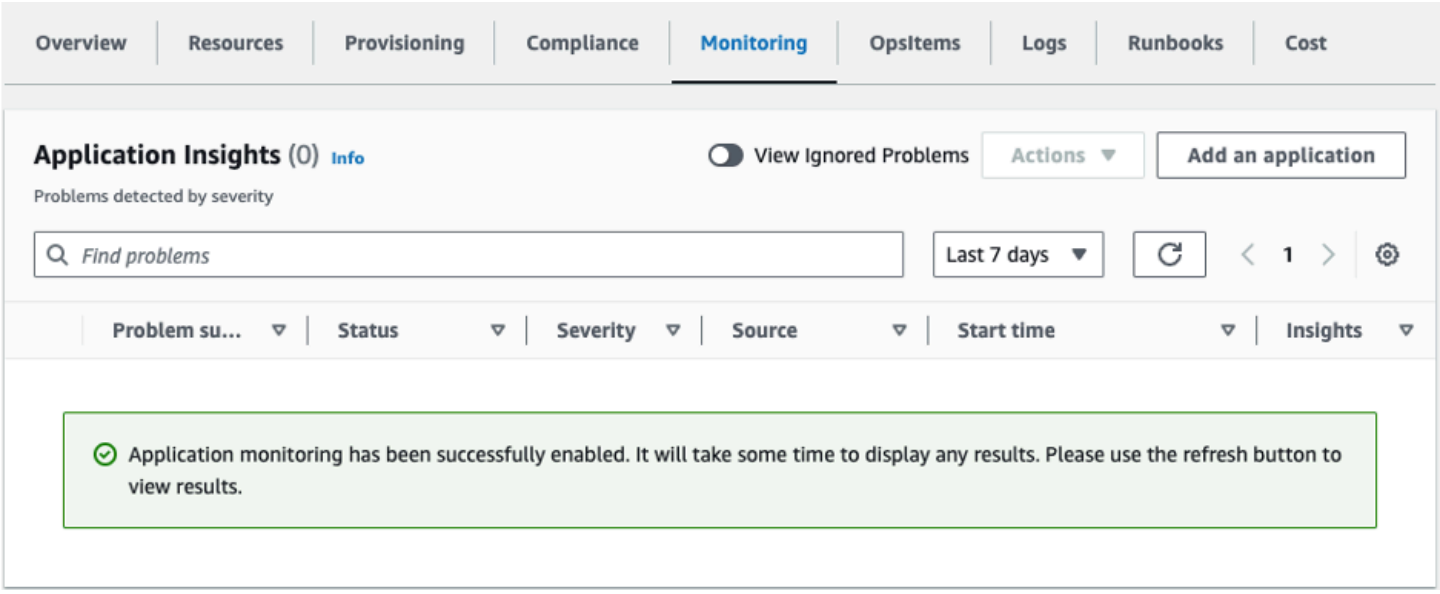
3. In **Applications**, search for the application name for this solution and select it.

   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**.



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

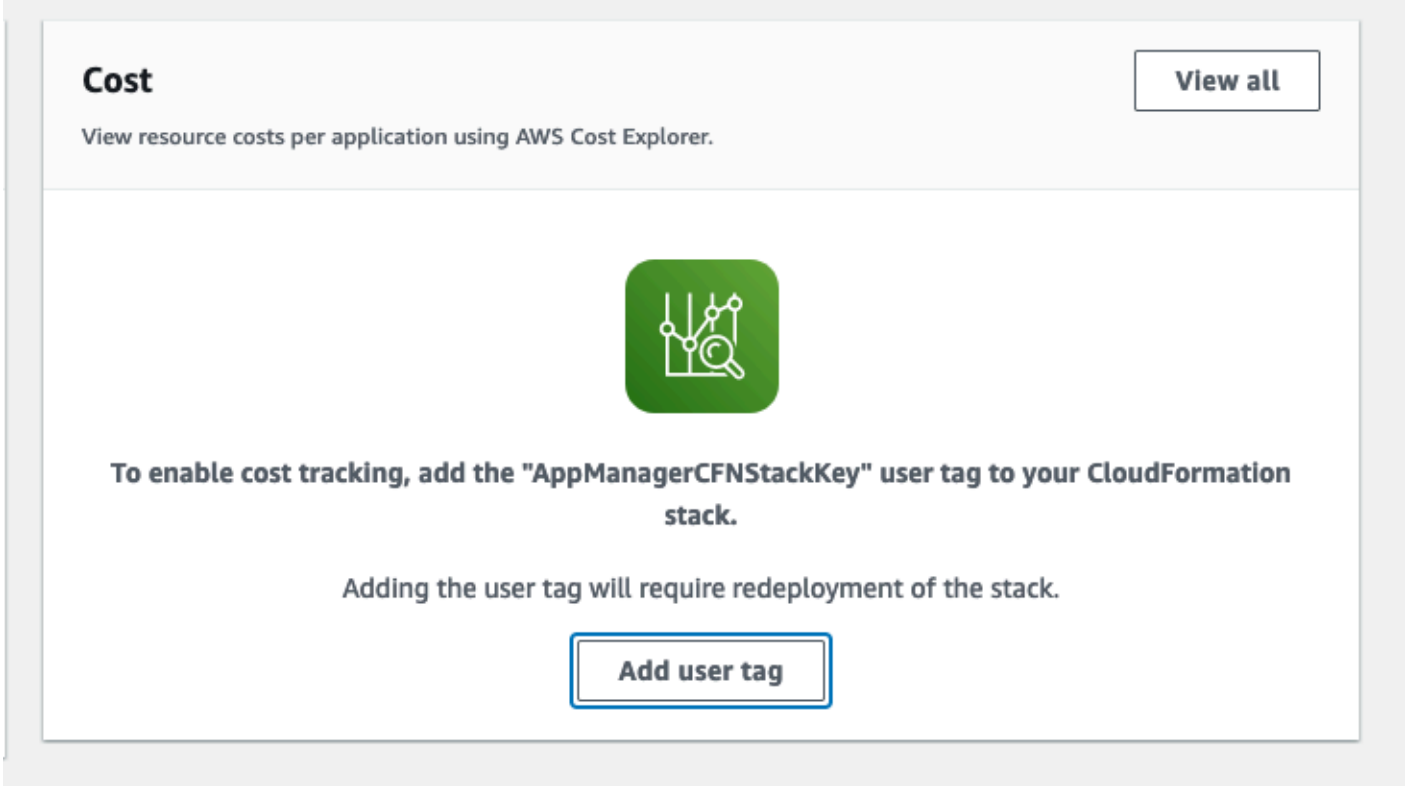


Confirm cost tags associated with the solution

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**.



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 the solution, follow this procedure to update the Maintaining Personalized Experiences with Machine Learning on AWS CloudFormation stack to get the latest version of the solution's framework.

1. Log in to AWS CloudFormation console, select your existing Maintaining Personalized Experiences with Machine Learning CloudFormation stack, and select Update. stack, and choose **Update**.
2. Select **Replace current template**.
3. Under **Specify template:**
   - Select Amazon S3 URL
   - Copy the link of the latest template.
   - Paste the link in the **Amazon S3 URL** box.
   - Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**. Choose **Next** again.
4. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1: Launch the stack for details about the parameters.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create (IAM) resources.
8. Choose **View change set** and verify the changes.
9. Choose **Update 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 status in approximately 15 minutes.

Troubleshooting

If you need help with this solution, contact AWS Support to open a support case for this solution.

Contact AWS 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 AWS 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 Maintaining Personalized Recommendations with Machine Learning solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the Personalize bucket and scheduler AWS DynamoDB table created by this solution. AWS Solutions Implementations do not automatically delete buckets and DynamoDB databases 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 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>
```

Deleting the Amazon S3 bucket

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 bucket referred to in the outputs of the stack.
4. Select the S3 bucket and choose **Delete**.

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

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

Deleting the scheduler Amazon DynamoDB database

The solution is configured to retain the scheduled items configured in the DynamoDB database if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can remove this database if it is not required. Follow these steps to delete it:

1. Sign in to the Amazon DynamoDB console.
2. Choose **Tables** from the left navigation pane.
3. Select the table referencing the stack that was deleted.
4. Choose **Delete table**.
5. Check **Delete all CloudWatch alarms for this table**.
6. Type the word **delete** to confirm deletion of the table.
7. Choose **Delete**.

User guide

To start using the Maintaining Personalized Experiences with Machine Learning on AWS solution, first determine the name of the Amazon S3 bucket deployed by the AWS solution. Record the `PersonalizeBucketName` value under the **outputs** tab of the CloudFormation stack deployed in Step 1. Launch the stack. You can use the AWS Management Console to copy configuration `0json` files to your `PersonalizeBucket` bucket.

By default, the AWS solution detects configuration files uploaded to subfolders under the **train** folder in the `PersonalizeBucket` deployed by the solution. Configuration files must be valid JSON and their file names must end in `0json`.

> **Note**
>
> When managing multiple dataset groups, you must use different subfolders for each dataset group's configuration. For example, for maintaining two dataset groups named **customer_1** and **customer_2**, you can create configuration files as follows:
>
> - `s3://<personalize_bucket_name>/train/customer_1/customer_1_config.json`
> - `s3://<personalize_bucket_name>/train/customer_2/customer_2_config.json`
>
> This allows you to keep the input datasets for each dataset group separate.

Creating dataset groups

Using a domain dataset group

The top-level container for datasets in Amazon Personalize is the dataset group. To create a domain dataset group (for E-commerce or video on demand use cases), upload one of the following configurations (changing the name to represent your use case):

E-commerce

```
{
    "datasetGroup": {
```

```
        "serviceConfig": {
            "name": "customer_1",
            "domain": "ECOMMERCE"
            "tags": [{"tagKey": "department", "tagValue": "finance"}]
        }
    }
}
```

Video on demand

```
{
    "datasetGroup": {
        "serviceConfig": {
            "name": "customer_1",
            "domain": "VIDEO_ON_DEMAND"
        }
    }
}
```

This configuration will create a dataset group with the domain "VIDEO_ON_DEMAND". Tags when specified under the "serviceConfig" of datasetGroup, only create tags for the datasetGroup. The same applies for any "serviceConfig" definition.

> **Note**
>
> As of March 2023, you can optionally provide tags for your resources in this solution. Tags can be provided at a time in "serviceConfig" definition of resources, or at root level (see sample below). Root level tags are applied to all resources in the configuration. Amazon Personalize allows tagging many resource types, such as dataset group, imports, recommenders, etc. Refer README.md in the github repository to view examples of using tagging resources.
> When not specified at root level, the location of tags in the config specifies which resource it is referring to.

Root-level tags

```
{
    "tags": [{"tagKey": "department", "tagValue": "finance"}],
    "datasetGroup": {
```

```
        "serviceConfig": {
            "name": "customer_1",
            "domain": "VIDEO_ON_DEMAND"
        }
    }
}
```

These tags will now be applied to all Amazon Personalize Resources mentioned in the configuration, such as "datasetGroup" in this example. If more resources were mentioned in this configuration, such as solution, campaigns, etc., it would be applied to all resources.

Using the CUSTOM dataset group domain

To create a custom dataset group omit the domain key as in the following configuration:

```
{
    "datasetGroup": {
        "serviceConfig": {
            "name": "customer_1"
            "tags": [{"tagKey": "department", "tagValue": "finance"}]
        }
    }
}
```

Note

- For both domain and custom dataset groups,* do not provide a `kmsKeyArn` or `roleArn` to the `datasetGroup.serviceConfig` path – if you provided a Personalize Key ARN in Step 1, the solution will automatically use the key provided.

Creating and maintaining datasets

Amazon Personalize solutions cannot be configured without data being present. To supply data to the Amazon Personalize service, you can upload data directly (through a dataset import job) and/or through an Amazon Personalize event tracker.

> **Note**
>
> Specifically, to tag a dataset import, the solution now supports an optional "serviceConfig" definition directly under the "datasets" field.

When using a domain dataset group, you must specify an identical domain for both the dataset group service config and each schema service config:

```
{
  "datasetGroup": {
    "serviceConfig": {
      "name": "customer_1_datasetgroup",
      "domain": "ECOMMERCE",
      "tags": [{"tagKey": "department", "tagValue": "finance"}]
    },
    "workflowConfig": {
      "schedules": {
        "import": "cron(0 */6 * * ? *)"
      }
    }
  },
  "datasets": {
    "serviceConfig": {
            "tags": [{"tagKey": "importNum", "tagValue": "finance-import-1"}]
        },
    "interactions": {
      "dataset": {
        "serviceConfig": {
          "name": "customer_1_interactions",
          "tags": [{"tagKey": "department", "tagValue": "finance"}]
        }
      },
      "schema": {
        "serviceConfig": {
          "name": "customer_1_interactions_schema",
          "domain": "ECOMMERCE",
          "schema": {
            "type": "record",
            "name": "interactions",
            "namespace": "com.amazonaws.personalize.schema",
            "fields": [
```

```
              {
                "name": "ITEM_ID",
                "type": "string"
              },
              {
                "name": "USER_ID",
                "type": "string"
              },
              {
                "name": "TIMESTAMP",
                "type": "long"
              },
              {
                "name": "EVENT_TYPE",
                "type": "string"
              },
              {
                "name": "EVENT_VALUE",
                "type": "float"
              }
            ]
          }
        }
      }
    }
  },
  "eventTracker": {
    "serviceConfig": {
      "name": "customer_1_event_tracker"
    }
  }
}
```

When using custom dataset groups, a domain is not provided to the dataset group or to the schema(s):

```
{
  "datasetGroup": {
    "serviceConfig": {
      "name": "customer_1_datasetgroup"
    },
    "workflowConfig": {
      "schedules": {
```

```
        "import": "cron(0 */6 * * ? *)"
      }
    }
  },
  "datasets": {
  "serviceConfig": {
            "tags": [{"tagKey": "importNum", "tagValue": "finance-import-1"}]
      },
    "interactions": {
      "dataset": {
        "serviceConfig": {
          "name": "customer_1_interactions"
        }
      },
      "schema": {
        "serviceConfig": {
          "name": "customer_1_interactions_schema",
          "schema": {
            "type": "record",
            "name": "interactions",
            "namespace": "com.amazonaws.personalize.schema",
            "fields": [
              {
                "name": "ITEM_ID",
                "type": "string"
              },
              {
                "name": "USER_ID",
                "type": "string"
              },
              {
                "name": "TIMESTAMP",
                "type": "long"
              },
              {
                "name": "EVENT_TYPE",
                "type": "string"
              },
              {
                "name": "EVENT_VALUE",
                "type": "float"
              }
            ]
          }
```

```
        }
      }
    }
  },
  "eventTracker": {
    "serviceConfig": {
      "name": "customer_1_event_tracker"
    }
  }
}
```

For more information on how to declare dataset schemas for Amazon Personalize, refer to Datasets and schemas, and follow the guidance for creating domain datasets and schemas or custom datasets and schemas.

To configure the dataset schema(s) required, you can update the configuration file to S3.

Maintaining Personalized Recommendations with Machine Learning inspects the paths `datasets.users.schema.serviceConfig`, `datasets.items.schema.serviceConfig` and `datasets.interactions.schema.serviceConfig` for dataset schema configuration. Schemas can be shared across dataset groups, but must be fully declared in the configuration file.

> **Note**
>
> Do not provide a `datasetGroupArn` or `schemaArn` to any `datasets.*.serviceConfig` path as they will be inferred by the solution.

The AWS solution imports data provided at specific paths relative to the uploaded configuration file. For example, if the configuration file is uploaded to `s3://<personalize_bucket_name>/train/customer_1/customer_1_config.json`, the AWS solution attempts to import data from the following paths (and will complete the import after data is found at one of those paths).

Data type	Path order to check
Interactions	`s3://<personalize_bucket_name>/train/customer_1/interactions/*.csv`

Data type	Path order to check
	`s3://<personalize_bucket_name>/train/customer_1/interactions.csv`

Users	`s3://<personalize_bucket_name>/train/customer_1/users/*.csv`
	
	`s3://<personalize_bucket_name>/train/customer_1/users.csv`

Items	`s3://<personalize_bucket_name>/train/customer_1/items/*.csv`
	
	`s3://<personalize_bucket_name>/train/customer_1/items.csv`

Note

If no import data is provided, the AWS solution continues to create resources. This allows you to create an event tracker to add items dynamically to your dataset group. Without data, the AWS solution fails to create an Amazon Personalize solution and solution version. To create an Amazon Personalize solution and solution version once sufficient data has been added to the dataset group, re-run the step function with the same input, or upload the configuration file again.

In both batch inference and batch segment scenarios, it is useful to re-run a dataset import job (for new data that was uploaded to the Personalize bucket, for example) on a schedule. To define a dataset import job schedule, add a cron schedule `datasetGroup.workflowConfig.schedules.import` path.

Creating and maintaining solutions

For custom datasets groups, an Amazon Personalize solution is the combination of an Amazon Personalize recipe, customized parameters and one or more solution versions (trained models). This AWS solution allows you to declare solution configuration using the same parameters as the

Amazon Personalize API, and set schedules for solution version retraining. To configure Maintaining Personalized Recommendations with Machine Learning, you can update the configuration file (the datasets section from the above configuration has been omitted for brevity, but are required):

> **Note**
>
> All configuration parameters for solution creation are supported. For information on additional parameters to supply when configuring an Amazon Personalize solution, use the parameters from the CreateSolution API documentation in the *Amazon Personalize Developer Guide*. The `datasetGroupARN` is inferred by the solution and must not be provided.

```
{
  "datasetGroup": {
    "serviceConfig": {
      "name": "customer_1_datasetgroup"
    },
    "workflowConfig": {
      "schedules": {
        "import": "cron(0 */6 * * ? *)"
      }
    }
  },
  "datasets": {
    "see": "above"
  },
  "solutions": [
    {
      "serviceConfig": {
        "name": "customer_1_user_personalization",
        "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
        "tags": [{"tagKey": "config", "tagValue": "config-011"}]
      },
      "workflowConfig": {
        "schedules": {
          "full": "cron(30 0 * * ? *)",
          "update": "cron(0 * * * ? *)"
        }
      },
      "campaigns": [
```

```
        {
          "serviceConfig": {
            "name": "customer_1_user_personalization_cpn",
            "minProvisionedTPS": 1,
            "tags": [{"tagKey": "department", "tagValue": "finance"}]
          }
        }
      ]
    }
  ]
}
```

When using the `aws-user-personalization` recipe, the service automatically updates the solution version in the background every 2 hours (at no additional cost). This auto-update process brings in new items added since the last update so that they can start being recommended to users. If the 2 hour auto-update is not frequent enough for introducing new items, you can have the Maintaining Personalized Experiences with Machine Learning solution create a new solution version on a cron schedule specified at the path `solutions[idx].workflowConfig.schedules.update`. An update retraining does not fully retrain the model. To occasionally create a new solution version (a full retraining to recalculate weights across the model based on all data), specify a cron schedule at the path `solutions[idx].workflowConfig.schedules.full`. As of September 2021, Amazon Personalize recommends training a new model weekly. The schedules specified in the example configuration have FULL training performed daily at 00:30 UTC and update training performed hourly.

Note

With User-Personalization, Amazon Personalize automatically updates the latest model (solution version) every two hours behind the scenes to include new data. There is no cost for these automatic updates. When solution versions are updated (via FULL or UPDATE training) the campaigns associated with the solution are updated to use the new solution version. This will result in a training cost to create a new solution version. Additonally, tags are optional. .

Scheduling batch inference jobs

To schedule batch inference jobs for your solutions, you must specify the `batchInferenceJobs` key under the solution with which you want to perform the batch inference.

```
{
  "datasetGroup": {
    "serviceConfig": {
      "name": "customer_1_datasetgroup"
    },
    "workflowConfig": {
      "schedules": {
        "import": "cron(0 */6 * * ? *)"
      }
    }
  },
  "datasets": { … }
  "solutions": [
    {
      "serviceConfig": {
        "name": "customer_1_user_personalization",
        "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization"
      },
      "workflowConfig": {
        "schedules": {
          "full": "cron(30 0 * * ? *)",
          "update": "cron(0 * * * ? *)"
        }
      },
      "campaigns": [
        {
          "serviceConfig": {
            "name": "customer_1_user_personalization_cpn",
            "minProvisionedTPS": 1
          }
        }
      ],
      "batchInferenceJobs": [
        {
          "serviceConfig": {
             "tags": [{"tagKey": "department", "tagValue": "finance"}]
          },
          "workflowConfig": {
```

```
            "schedule": "cron(0 3 * * ? *)"
          }
        }
      ]
    }
  ]
}
```

> **Note**
>
> The batch inference jobs created use the most recent solution version discovered for their solution. Additional batch inference job parameters can be specified under the `serviceConfig` key (empty above). The `jobInput`, `jobName`, `jobOutput`, `roleArn`, and `solutionVersionArn` are inferred by the Maintaining Personalized Experiences with Machine Learning solution.

The solution expects batch inference job input .json file in an Amazon S3 location at the following path:

```
s3://<personalize_bucket_name>/batch/<dataset_group_name>/<solution_name>/
job_config.json
```

The solution outputs batch inference job results in an Amazon S3 location at the following path:

```
s3://<personalize_bucket_name>/batch/<dataset_group_name>/<solution_name>/<job_name>/*
```

This example cron configuration above performs a batch inference job nightly at 03:00 UTC.

Scheduling batch segment jobs

To schedule batch segment jobs for your solutions, you must specify the `batchSegmentJobs` key under the solution with which you want to perform the batch segment job.

```
{
  "datasetGroup": {
    "serviceConfig": {
      "name": "customer_1_datasetgroup"
    },
```

```
    "workflowConfig": {
      "schedules": {
        "import": "cron(0 */6 * * ? *)"
      }
    }
  },
  "datasets": { … }
  "solutions": [
    {
      "serviceConfig": {
        "name": "customer_1_item_affinity",
        "recipeArn": "arn:aws:personalize:::recipe/aws-item-affinity"
      },
      "workflowConfig": {
        "schedules": {
          "full": "cron(30 0 * * ? *)",
          "update": "cron(0 * * * ? *)"
        }
      },
      "batchSegmentJobs": [
        {
          "serviceConfig": {},
          "workflowConfig": {
            "schedule": "cron(0 3 * * ? *)"
          }
        }
      ]
    }
  ]
}
```

Note

The batch segment jobs created use the most recent solution version discovered for their solution. Additional batch segment job parameters can be specified under the `serviceConfig` key (empty above). The `jobInput`, `jobName`, `jobOutput`, `roleArn`, and `solutionVersionArn` are inferred by the Maintaining Personalized Experiences with Machine Learning solution.

The solution expects batch inference job input `0json` file in an Amazon S3 location at the following path:

```
s3://<personalize_bucket_name>/batch/<dataset_group_name>/<solution_name>/
job_config.json
```

The solution outputs batch inference job results in an Amazon S3 location at the following path:

```
s3://<personalize_bucket_name>/batch/<dataset_group_name>/<solution_name>/<job_name>/*
```

This example cron configuration above performs a batch segment job nightly at 03:00 UTC.

Receiving job status notifications

You can configure Amazon EventBridge or CloudWatch Events to notify you with status updates for ongoing workflows, such as importing data, generating new solution versions, and creating batch inference jobs. EventBridge and CloudWatch Events deliver a near real-time stream of events that describe changes in AWS resources. For example, you can set up an event to notify you when an Amazon Personalize batch inference job finishes and notifies a third party system with the batch inference job results.

Events are emitted on a best-effort basis. For more information about events, refer to the Amazon EventBridge User Guide and the Amazon CloudWatch Events User Guide.

> **Note**
>
> We recommend using Amazon EventBridge to manage events. CloudWatch Events and EventBridge use the same API and provide the same functionality, but EventBridge provides more features. Changes that you make in either CloudWatch or EventBridge will appear in each console. For more information, refer to the Amazon EventBridge documentation.

Monitoring personalize workflows

An event indicates a change in your AWS environment, and a rule matches incoming events and routes them to targets for processing. You can set up rules to match events generated by this solution and route them to one or more target functions or streams. EventBridge and CloudWatch Events detect events as they occur and invoke the target in the matching rule.

The following table lists resources and status change events that you can monitor:

Resource	Status change event name	Status
Dataset Group	Personalize Dataset Group State Change	ACTIVE, CREATE IN_PROGRESS
Dataset	Personalize Dataset State Change	ACTIVE, CREATE IN_PROGRESS
Dataset Import Job	Personalize Dataset Import Job State Change	ACTIVE, CREATE IN_PROGRESS
Recommender	Personalize Recommender State Change	ACTIVE, CREATE IN_PROGRESS
Solution	Personalize Solution State Change	ACTIVE, CREATE IN_PROGRESS
Solution Version	Personalize Solution Version State Change	ACTIVE, CREATE IN_PROGRESS
Campaign	Personalize Campaign State Change	ACTIVE, CREATE IN_PROGRESS, UPDATING
Batch Inference Job	Personalize Batch Inference Job State Change	ACTIVE, CREATE IN_PROGRESS
Batch Segment Job	Personalize Batch Segment Job State Change	ACTIVE, CREATE IN_PROGRESS

Notifications contain information about the resource, including the Amazon Resource Name (ARN), job status, job duration (in seconds), and, if the job failed, an error message. The following code snippet is an example notification:

```
{
    "version": "0",
    "id": "345a0b30-35d5-4c9f-89e4-1fc1200ee77e",
    "detail-type": "Personalize Solution Version State Change",
    "source": "solutions.aws.personalize",
    "account": "111122223333",
```

```
    "time": "2021-10-31T01:19:15Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:personalize:us-east-1:111122223333:solution/
user_personalization_solution/aaaaaaaa"
    ],
    "detail": {
        "Arn": "arn:aws:personalize:us-east-1:111122223333:solution/
user_personalization_solution/aaaaaaaa",
        "Status": "ACTIVE",
        "Duration": 1116
    }
}
```

Creating an EventBridge Rule for job status notifications

You can create an EventBridge rule to notify you of the status changes for ongoing workflows in two ways:

1. For the AWS Console, you can refer to Creating Amazon EventBridge rules that react to events in the *Amazon EventBridge User Guide* for more information.

   In the sample procedure, for Service provider, select **Custom pattern**, and configure your event pattern per the Amazon EventBridge User Guide

   For example, add the following parameters to start a rule when any batch inference job is completed:

   ```
   {
       "source": ["solutions.aws.personalize"],
       "detail-type": ["Personalize Batch Inference Job State Change"],
       "detail": {
           "Status": ["ACTIVE"]
       }
   }
   ```

2. To modify the solution, after cloning the repository from GitHub, edit this python file. Refer the Boto3 Documentation on EventBridge for details on parameters and its syntax.

Importing existing resources

It is useful to establish schedules for resources that already exist within Amazon Personalize (for example, to schedule FULL solution version retraining). To accomplish this, you can use the included scheduler CLI.

Installing the scheduler CLI

The scheduler CLI is available with the solution source code, under the `source` directory. Use the following commands to install the CLI. It is recommended that you use a separate virtual environment for this installation.

```
cd source
pip install --upgrade pip
pip install cdk_solution_helper_py/helpers_common
pip install scheduler/common
```

Using the scheduler CLI

You can use the included help commands to list command options:

```
> aws-solutions-scheduler --help

Usage: aws-solutions-scheduler [OPTIONS] COMMAND [ARGS]...

  Schedluer CLI

Options:
  -s, --stack TEXT                [required]
  -r, --region TEXT               [required]
  --scheduler-table-name-output TEXT
  --scheduler-stepfunction-arn-output TEXT
  --help                          Show this message and exit.

Commands:
  activate              Activate a scheduled task
  deactivate            Deactivate a scheduled task
  describe              Describe a scheduled task
  import-dataset-group  Create a new configuration from an existing...
  list                  List all scheduled tasks
```

Listing active schedules

```
> aws-solutions-scheduler -s <stack_name> -r <region_name> list
{
  "tasks": [
    "personalize-dataset-import-item-recommender",
    "solution-maintenance-full-item-recommender-user-personalization"
  ]
}
```

Describing a scheduled task

Describing a scheduled task will show you its name, version, status (activated or deactivated), schedule, and target step function:

```
> aws-solutions-scheduler -s <stack_name> -r <region_name> describe –task
{
  "task": {
    "active": true,
    "name": "personalize-dataset-import-item-recommender",
    "schedule": "cron(*/15 * * * ? *)",
    "step_function": "arn:aws:states:us-east-1:
 111122223333:stateMachine:personalizestack-periodic-dataset-import-aaaaaaaaaaaa",
    "version": "v1"
  }
}
```

Deactivating tasks

Deactivating a task will stop any active schedules but retain the scheduled task for future reactivation.

```
> aws-solutions-scheduler -s PersonalizeStack -r us-east-1 deactivate --task solution-
maintenance-full-item-recommender-user-personalization
{
  "task": {
    "active": false,
    "name": "solution-maintenance-full-item-recommender-user-personalization",
    "schedule": "cron(0 */6 * * ? *)",
```

```
    "step_function": "arn:aws:states:us-east-1:
 111122223333:stateMachine:personalizestack-periodic-solution-maintenance-
aaaaaaaaaaaa",
    "version": "v1"
  }
}
```

Activating tasks

Activating a task will ensure that the scheduler step function is running for the task on the specified schedule.

```
> aws-solutions-scheduler -s PersonalizeStack -r us-east-1 activate --task solution-
maintenance-item-recommender-user-personalization
{
  "task": {
    "active": true,
    "name": "solution-maintenance-full-item-recommender-user-personalization",
    "schedule": "cron(0 */6 * * ? *)",
    "step_function": "arn:aws:states:us-east-1:
 111122223333:stateMachine:personalizestack-periodic-solution-maintenance-
aaaaaaaaaaaa",
    "version": "v1"
  }
}
```

Importing an existing resource and setting schedules

It can be useful to add scheduling to resources that already exist in Amazon Personalize but that are not managed by the scheduler or the solution. Use the following code to import and set schedules:

```
aws-solutions-scheduler -s <stack_name> -r <region_name> import-dataset-group
 -d <dataset_group_name> -i "cron(*/15 * * * ? *)" -f "item-recommender-user-
personalization@cron(0 */6 * * ? *)" -p train/item-recommender/config.json
```

The CLI supports cron-style expressions for import and solution version UPDATE and FULL training.

Note

While the solution supports batch inference job scheduling, the scheduler CLI cannot establish schedules for batch import. To establish a schedule for batch import, first import the dataset group using the CLI, then follow the instructions in Scheduling batch inference jobs.

KMS customer-managed key policy

The following is a valid key policy for use with the solution when using the Personalize KMS Key ARN parameter.

Warning

If KMS is required, ensure the Personalize KMS Key ARN is provided when the stack is created. Do not update the stack to allow KMS. Do not remove the key after it is set. Doing either will result in resources being unable to update due to different security configurations.

```
{
    "Version": "2012-10-17",
    "Id": "PersonalizePolicy",
    "Statement": [
        {
            "Sid": "Allow use for the Personalize service",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<account_id>:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

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 Maintaining Personalized Experiences with Machine Learning templates are generated using the AWS Cloud Development Kit (AWS CDK) (AWS CDK). Refer to the README.md file for additional information.

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 Maintaining Personalized Recommendations with Machine Learning deployment
- **Timestamp** - Data-collection timestamp

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

1. Download the `maintaining-personalized-experiences-with-machine-learning.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:

```
AnonymousData:
    SendAnonymousData:
      Data: Yes
```

   to:

```
AnonymousData:
    SendAnonymousData:
      Data: No
```

4. Sign in to the AWS 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 Launch the stack in the Automated Deployment section of this guide.

Contributors

- Supreet Kaur Takkar
- Paul Shuparski-Miller

Revisions

Check the CHANGELOG.md file in the GitHub repository to see all notable changes and updates to the software. The changelog provides a clear record of improvements and fixes for each version.

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.