Implementation Guide

Unified Profiles for Travelers and Guests on AWS

Unified Profiles for Travelers and Guests on AWS: Implementation Guide

Table of Contents

Create a 360-degree view of travelers and guests to curate key insights in real time

Publication date: *August 2023 ([last update](): February 2024)*

The Unified Profiles for Travelers and Guests on AWS solution connects customer data to unlock insights that drive personalization, retailing, and curated experiences for travelers and guests.

Building a 360-degree view of the traveler and guest experience is a top priority for most travel brands; however, stitching together disparate traveler data from siloed systems often prevents this goal. Unified Profiles for Travelers and Guests on AWS was built to simplify the process of creating this 360-degree view of travelers to allow travel brands key insights into the journey of travelers and guests in real time.

This implementation guide provides an overview of the solution, its reference architecture and components, considerations for planning the deployment, configuration steps for deploying the solution to the Amazon Web Services (AWS) Cloud.

The intended audience for discovering and 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. The estimated cost for running this solution in the US East (N. Virginia) Region is USD $0.038 per traveler profile per year.	[Cost]()
Understand the security considerations for this solution.	[Security]()

If you want to . . .	Read . . .
Note This solution stores Travelers Personal Identifiable Information (PII).	
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 CloudFormation 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:

Ingestion

The solution allows AWS customers to ingest traveler data in batch and real time. The solution integrates with Travel and Hospitality Application Connectors Catalog on AWS to provide off-the-shelf data feeds from eleven hotel property management systems (PMS) and clickstream events from Tealium Customer Data Platform (CDP).

Transformation

The solution transforms incoming traveler data using industry-specific semantic logic to preserve as much identity information as possible. Any error occurring during the ingestion process is queued and stored in an Amazon DynamoDB database for further processing.

Identity resolution

The solution provides rule-based and AI-based identity resolution capabilities. You can configure the rules in the Amazon Connect Customer Profiles console page. The solution's modular

architecture can also be extended to support custom-matching logic and use AWS partners, such as [Amperity](#).

Secure storage and permission management

Sensitive traveler data is stored in AWS specialized customer data storage (Amazon Connect Customer Profiles). The solution provides a granular permission system allowing you to implement detailed access control. This solution creates roles with least privileges, and these roles grant the solution's resources the needed permissions. This feature allows you to provide only the strictly necessary access to traveler data at the time it is needed.

Analytics and traveler data visualization

The solution allows customers to query and analyze the sanitized data and merged profiles using Amazon Athena and to build [Amazon QuickSight](#) or [Tableau dashboards](#) in minutes. (See Querying the data using Amazon Athena for more details)

Real-time traveler change capture

The solution provides a real-time feed of its profile data store. This allow you to react in real time to changes in traveler profiles (such as booking cancellations or hotel checkouts) and leverage the 360-degree view of the traveler to engage them using personalized communications.

Operational monitoring

The solution provides a dedicated [Amazon CloudWatch dashboard](#) template allowing you to configure for monitoring data ingestion and profile matching operations, and to set up alerts.

Extensible design

The solution is built on the concept of generic business objects, which makes it easy to extend to additional data such as contact center logs or traveler reviews.

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

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

Use cases

Traveler journey personalization

By reacting in real time to changes in traveler's profiles, you can send personalize messages to the traveler at key steps of the travel journey. Access to the full 360-degree view of the traveler allows personalization of messages along with potential upsell and cross-sell recommendations.

Disruption handling

During times of operational difficulties (such as severe weather or technology issues) leading to mass changes and cancellation, you can leverage the real-time profile feed to engage with travelers strategically with personalized messages and offers.

Traveler personas and segmentation

The solution outputs all profiles in real time to an Amazon Simple Storage Service (Amazon S3) bucket. Data analysts can create advanced queries to support definitions of traveler segments and personas using Amazon Athena. This capability allows you to mitigate the frequent blur between traveler segments and the need of hybrid segments (bleisure) by using a more dynamic approach to traveler segmentation.

Personalized and automated customer service

The solution provides an out-of-the-box user interface allowing customer service agents with appropriate permissions to access traveler information in real time and summarize in a 360-degree view that can be configured with dynamic links to systems of records such as PMS, customer record system (CRS), and passenger service systems (PSS). This allows customer service agents to provide a personalized service while increasing their productivity. The data is also available through an API allowing you to build your own dashboards or integrate the Traveler 360 profile in your in-house customer service application and contact center (either Amazon Connect or another solution).

Concepts and definitions

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

Travel business object

The solution uses business objects (booking, clickstream event, hotel stay, loyalty profile…) as inputs to the solution. Published AWS travel and hospitality schema defines the JSON document

format (see [Sending data to the real-time stream](#) section). These objects are similar in format to the source system format and are being transformed downstream into Amazon Connect Customer Profile records.

Amazon Connect Customer Profiles records

The records are a result of the semantic transformation of a travel business object into an identity-preserving flat record that can be ingested into Amazon Connect Customer Profile records (our profile data store).

Identity resolution

The process of identifying (and potentially merging) duplicate traveler profiles based on configured rules or AI.

> ⓘ **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.
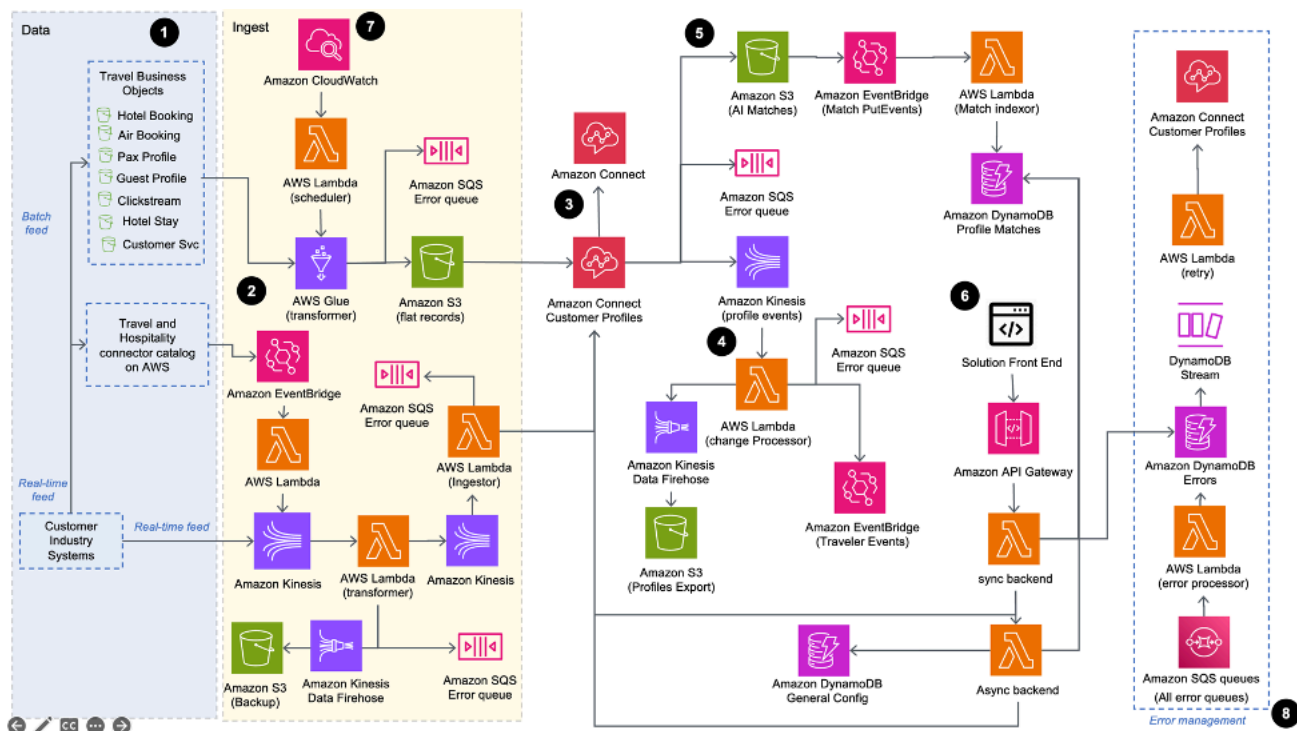
Unified Profiles for Travelers and Guests on AWS architecture

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

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

1. **Data** – The solution allows customers to ingest data using a batch feed or a real-time feed. The batch feed has the option to ingest data directly into the solution's Amazon S3 bucket or use the AWS travel and hospitality connector side-car solution. The real-time feed ingests data by sending objects directly to Amazon Kinesis. Data must follow the published Travel and Hospitality Application Connecters Catalog on AWS schema. All objects sent to the feed are backed up in Amazon S3 using Amazon Data Firehose.

2. **Ingest** - Batch feed data is processed using AWS Glue. Real time data is processed using AWS Lambda. The data is transformed into Amazon Connect Customer Profiles records, a flat format suitable for ingestion into Amazon Connect Customer Profiles.

3. Data is ingested in Amazon Connect Customer Profiles. Real-time matching rules are applied along with weekly machine learning-based identity resolution jobs.

4. Amazon Connect Customer Profiles exports profile changes in real time to a Kinesis data stream. A Lambda function retrieves the full profile and ingests via Firehose into Amazon S3. Profiles are also published to Amazon EventBridge.

5. Additionally or instead of AWS Identity resolution, customers can use partners like Amperity or their own identity resolution logic to feed profile matches to Amazon DynamoDB.

6. A frontend allows data administrator to monitor ingestions and data quality. It also contain out-of-the-box profile search and display functions which are usable by customer-facing staff. Profiles can be manually merged in by users. Data access is managed by permissions using Amazon Cognito groups.

7. A CloudWatch-initiated Lambda function runs hourly to optimize data partitioning.

8. All ingestions and processing errors are recorded in a DynamoDB table via multiple Amazon Simple Queue Service (Amazon SQS) queues.

AWS Well-Architected design considerations

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

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the [operational excellence pillar](#).

This solution comes with an Amazon CloudWatch dashboard allowing customers to continuously monitor the health of the ingestion processes and set alarms to detect abnormal behaviors. Additionally, the solution provides a dedicated frontend page with aggregated metrics and status.

Security

This section describes how we architected this solution using the principles and best practices of the [security pillar](#).

All data ingesting in the solution is encrypted at rest and in transit. Every compute component is given strict least privileged IAM permissions. For details, refer to [IAM roles](#). The solution provides an advanced permission feature allowing the Administrator to enforce least privilege for users using functional rules. The solution has been reviewed and evaluated using the same strict process as AWS services and complies with the highest standard of security.

Reliability

This section describes how we architected this solution using the principles and best practices of the [reliability pillar](#).

The solution is an entirely serverless build using event-driven architectures. Nine error queues are plugged in on nine different steps of the ingestion process to catch any potential data ingestion issues. The error and source data is then stored in a DynamoDB table to be further processed. The solution retry logic implements a tradeoff between the SDK retries, Lambda retries of Kinesis data stream, and Amazon S3 Put events to ensure an optimal chance of successful real-time data ingestion.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the [performance efficiency pillar](#).

The solution's batch ingestion process only runs on new data added since the last run. The solution monitors the ingestion performance using a dedicated CloudWatch dashboard and monitors the response time of every API-initialized use case.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the [cost optimization pillar](#).

The solution exclusively uses serverless services, so you are charged based on use. The solution has been load-tested and profiled for cost, and had multiple cost optimization measures based on using data from production-like traffic simulation. Additional recommendations are provided in this Implementation Guide to allow customers to further optimize the cost of this solution based on their specific use.

Sustainability

This section describes how we architected this solution using the principles and best practices of the [sustainability pillar](#).

The solution uses managed and serverless services to minimize the environmental impact of the backend services. The solution's serverless design and the use of managed services are aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

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

Amazon Connect Customer Profiles domains

The data is ingested and stored in Amazon Connect Customer Profiles under one of several domains. Domains can be created using the solution's web frontend. During the creation of a domain, the solution creates multiple AWS resources (Amazon Connect mappings, AWS Glue tables, Amazon AppFlow flows, Amazon Cognito groups) allowing the domain to be fully functional and ready to receive data in seconds.

Batch ingestion

The solution batch ingestion process allows you to perform either one time or recurrent data ingestion from Amazon S3. The data is expected to be stored in the Amazon S3 bucket created by the solution (the bucket name can be found under the CloudFormation console **Output** tab). The data must be stored under a specific folder structure and the business objects must follow our travel and hospitality schema. Both folder structure and schema files are found in the Use the solution section.

Real-time ingestion

The solution allows customers to send real-time traveler data via Kinesis Data Streams. The data is added to the traveler profile in real time (within seconds) using sequence of identity resolution features detailed in this document. Messages sent to the Kinesis data stream created by the solution are expected to follow the AWS travel and hospitality schema. All data sent to the Kinesis data stream is backed-up in Amazon S3 via Firehose.

Travel and Hospitality Catalog on AWS connector integration

The solution integrates with the Travel and Hospitality Connector Catalog on AWS solution. Providing the name of the Amazon S3 bucket created by Travel and Hospitality Connector Catalog

on AWS (as input to the CloudFormation template of this solution) creates a real-time feed allowing data ingestion by the Travel and Hospitality Connector Catalog on AWS to be added in real-time to the traveler profile.

Real-time profile export

As traveler profiles are created, updated, and merged, a real-time stream is created out of Amazon Connect Customer Profiles allowing customers to react in real-time to any change within a profile. These events are processed by a Lambda function and stored in an Amazon S3 bucket. If the **eventbridgeActivated** template parameter is set to `true` (it is `true` by default), each event is also published to an EventBridge event bus prefixed `ucp-traveller-changes-`. This provides easy access to profile events for downstream use cases.

Identity resolution

The solution attempts to resolve the identity of the traveler during every datapoint provided in both real time and batch. The solution searches for a traveler ID field provided in the data (this is in case the customer already has identified the traveler and can provide the customer ID as an input within the solution). If no ID is provided, the solution will generate one then applies a set of deterministic matching rules. These rules can be configured within the AWS Management Console for Amazon Connect Customer Profiles.

An AI-based identity-resolution process runs weekly to identify profiles that might be duplicates. The process outputs AI matches to an Amazon S3 bucket. A Lambda function processes CloudWatch S3 Put events and indexes AI-matches in DynamoDB. Customers can choose to implement their own identity resolution (or that of a partner, such as Amperity) on top of or instead of the one provided with this solution. More details can be found in the Integrating third-party identity resolution section.

Profile ID match database

The non-deterministic identity resolution process feeds all potential duplicates with a match score to a DynamoDB table. The content of this database is accessed by the solution user interface allowing operators to manually merge these profiles. Additionally, you can subscribe to the DynamoDB stream allowing the solution to receive notifications when new matches are added for further offline process.

Error management queues

Any error or unexpected behavior occurring during both batch and real-time ingestion processes are fed to nine Amazon SQS queues in real time. Errors are then ingested and stored temporarily into a DynamoDB table and then moved long term into an Amazon S3 bucket. This allows you to have real-time insights on ingestion issues and potentially reprocess the data. Additionally, DynamoDB Streams is used to analyze all errors and retry specific ones.

Partition synchronization

A CloudWatch-triggered lambda function runs hourly to optimize data partitioning and synchronizes AWS Glue partitions for every domain. This allows close control over the partitioning and AWS Glue job run (as opposed to scheduled AWS Glue crawler) and allows for further customization and cost optimizations.

User interface

A web frontend is provided with this solution to allow users to easily monitor and manage the data process from a single screen and to search, retrieve, and merge traveler profiles.

AWS services in this solution

AWS service	Description
Amazon CloudFront	**Core.** Publishes the solution frontend static resources.
Amazon Connect Customer Profiles	**Core.** Stores, searches, retrieves, and merges the Traveler 360 profiles and performs rule-based and AI-based identity resolution.
AWS DataSync	**Core.** Sets up a one-time data transfer from the Travel and Hospitality Application Connectors Catalog on AWS S3 bucket to this solution's entry bucket allowing data processing from data ingested prior to the solution deployment.

AWS service	Description
Amazon DynamoDB	**Core.** Stores temporary configuration and identity resolution matches in a way that can be easily queried for this solution.
Amazon EventBridge	**Core.** Processes Amazon S3 Put events from the Travel and Hospitality Applicati on Connectors Catalog on AWS solution. A dedicated event bus is created by the solution and it receives events in the event of traveler data changes.
AWS Glue	**Core.** Ingests business objects in batch from Amazon S3.
Amazon Kinesis	**Core.** Ingests business objects in real time.
AWS Lambda	**Core**. The primary form of compute for this solution. It processes and transforms business objects in real time, processes ingestion errors, manages the AWS Glue table partitions, and processes profile changes in real time.
Amazon S3	**Core.** Intermediary storage for incoming business objects and profile matches. The solution subscribes to Amazon S3 PutObject events to process business object from the Travel and Hospitality Application Connector s Catalog solution and for processing identity resolution matches in real time.
Amazon SQS	**Core.** Stores any processing errors during the ingestion process.
Amazon CloudWatch	**Supporting.** Stores the solution logs and provides monitoring capabilities to customers with a premade CloudWatch dashboard.

AWS service	Description
Amazon Cognito	**Supporting.** Secures frontend access and manages users' granular permissions to access traveler profile data.

How this solution works

The solution reacts to traveler events across channels allowing gradual augmentation of the Traveler 360 profile in real time. For every event, the solution applies identity resolution logic in four steps.

1. The solution searches a traveler ID in the data. This allows the you to provide the identity of the traveler they have already resolved by sending the traveler ID with the relevant business object.

2. The solution applies a set of identity resolution deterministic matching rules configured in the Amazon Connect Customer Profiles console.

3. If no match is found, a new traveler ID is created in the database. If a match is found, the data is appended to the found profile.

4. On a weekly basis, an AI-powered identity resolution batch job runs through all the profiles and identifies the potential matches that could not be detected by the rules. Customer can then use the provided frontend to manually compare and merge profiles based on their content and match score.

All changes within the profile database are streamed to an Amazon Kinesis data stream. You can then choose to ingest the data into Amazon S3 for analytics purposes and send events to Amazon EventBridge allowing downstream real-time integrations (such as marketing automation software).

The following diagram provides an example of this process.
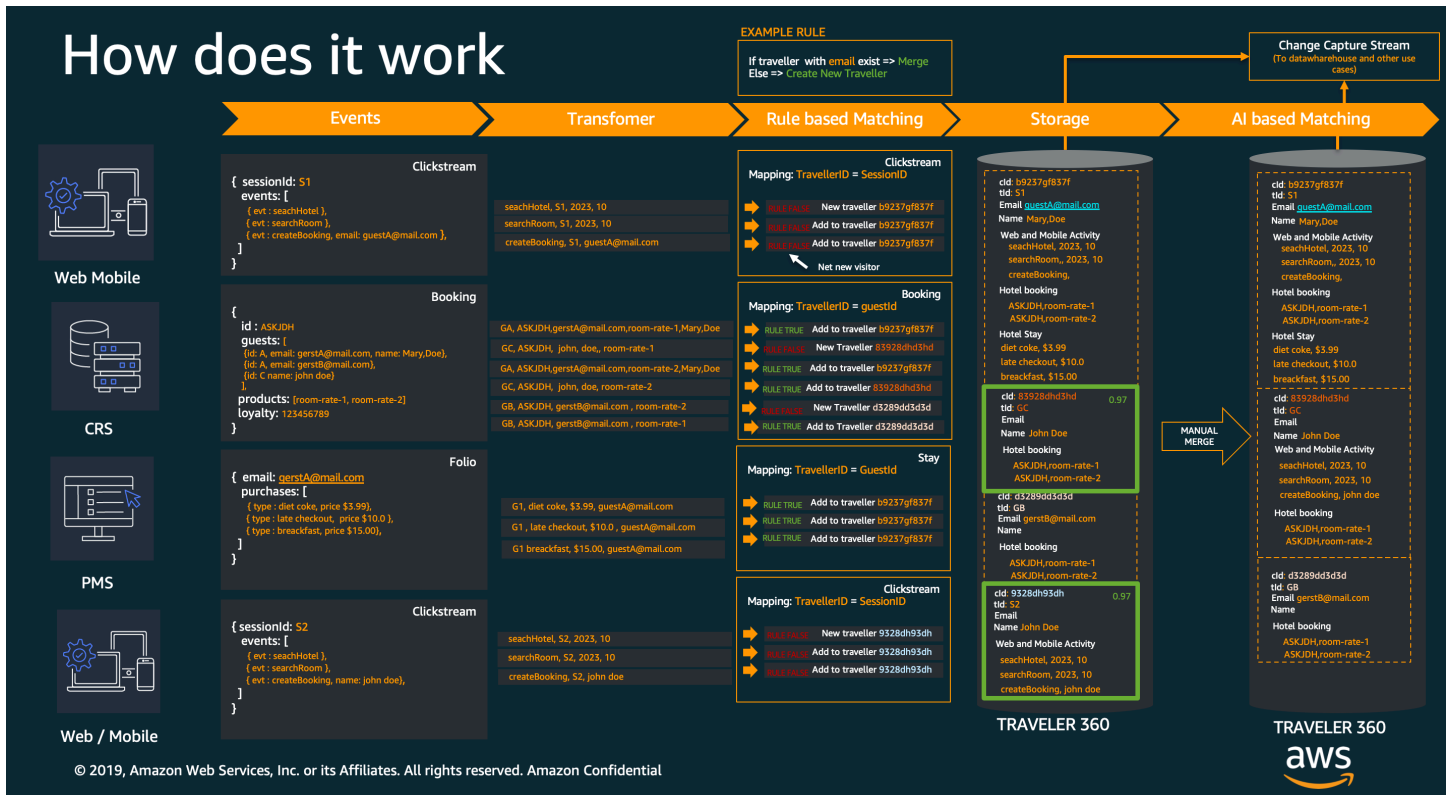
Example of real-time traveler data ingestion flow during the guest journey

Plan your deployment

This section describes the [cost](), [security](), [quota](), and other considerations prior to deploying the solution.

Supported AWS Regions

Unified Profiles for Travelers and Guests on AWS is supported in the following AWS Regions:

Region name	
US East (N. Virginia)	Asia Pacific (Tokyo)
US West (Oregon)	Canada (Central)
Asia Pacific (Singapore)	Europe (Frankfurt)
Asia Pacific (Seoul)	Europe (London)
Asia Pacific (Sydney)	

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 US East (N. Virginia) is approximately $0.038 per traveler profile per year.

> ⓘ **Note**
>
> The cost of this solution is proportional to the data ingested and therefore to the value provided by the solution.

We recommend creating a [budget]() through [AWS Cost Explorer]() to help manage costs. Prices are subject to change. For full details, see the pricing webpage for each AWS service used in this solution.

Sample cost table

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

Number of profiles	Cost per year [USD]
100,000 profiles	$3,800.00
1,000,000 profiles	$38,000.00
10,000,000 profiles	$380,000.00
Greater than 10,000,000 profiles	Contact your account team to discuss tiered pricing options.

> ⓘ **Note**
>
> 95% of the cost associated with this solution relates to Amazon Connect Customer Profiles. The cost estimates assume that 80% of profiles have 100 business objects or less associated.

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

AWS 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.

Amazon CloudFront

This solution deploys a web console [hosted](#) in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes a CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to the solution's website bucket contents. For more information, see [Restricting Access to Amazon S3 Content by Using an Origin Access Identity](#) in the *Amazon CloudFront Developer Guide*.

Amazon Cognito

The solution uses Amazon Cognito to secure access to the web frontend and configure and enforce permissions regarding traveler profile access. Customers can create granular permissions as Amazon Cognito groups. Users associated with these groups will only have access to the profile data specified in the group description.

The permission syntax is described in the section below:

Setting up permissions for Amazon Cognito

Every request to the Unified Customer Profile for Travelers and Guests on AWS main Lambda function goes through a set of standard processing steps (validation, processing, and post initiation).

One of the tasks of the post-processing step includes filtering all traveler data based on the end-user's permission. Filters can be applied to traveler data, as well as each individual business object.

Permissions are built using role-based access, and managed with Amazon Cognito user groups at the domain level. By default, users don't have access to any traveler data. Each new Unified Customer Profile domain has an admin group created by default for convenience. All other groups must be manually created for security and privacy.

Follow the steps below to create additional groups:

1. Navigate to the [Amazon Cognito console](#).
2. Find the user pool for the solution. The pool name starts with ucpUserpool.
3. Create a new user group for the new role with the following conventions:
   a. Name must have the following structure:

```
ucp-[domain name]+<role-name>
```

The domain name can be found from the solution user interface. The role name can be any text compliant with Amazon Cognito group name rules that describes the role purpose.

The role description must contain a string pattern that describes the permissions. Below are a few examples of the permission pattern:

- Admin user with full access:

```
*/*
```

- User with access to all traveler data and to air bookings with BOS as the departure or arrival airport:

```
traveller/*
      air_booking/*?from = BOS or to = BOS
```

- User with access to traveler data except the date of birth field. The user can also access hotel bookings where hotel_code is in a given list of values:

```
traveller/*,dob
hotel_booking/*?hotel_code in [BOS1, BOS2, BOS3]
```

4. Assign users to groups. The permission will take effect immediately

Additional security recommendations

We recommend the following security-related configurations in your AWS account for deploying this solution:

- Ensure that [AWS CloudTrail](#) is activated in your account. In case of security incident, CloudTrail can be a key tool for forensic analysis.
- The solution frontend is hosted on CloudFront and using the default URL created by CloudFront. We recommend you setup a custom domain with a dedicated certificate in order to impose the TLS version that meets your security team's requirement.

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.

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.

Deploy the solution

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

Deployment process overview

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

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: Create users

- Create users and groups and locate the UserPool.
- Create an Amazon Cognito user in this user pool for your administrator.

Step 3: Create an Amazon Connect Customer Profiles domain

- Create a domain name.

Step 4: Provide user permissions

- Locate permission group and assign permissions.

Step 5: Map the customer profile domain to your Amazon Connect instance

- Associate Amazon Connect Customer Profiles domain to instance.

Step 6: Create match and merge rules

- Match and merge rules for your domain.

> ⚠️ **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

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

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

This AWS CloudFormation template deploys **Unified Profiles for Travelers and Guest on AWS** in the AWS Cloud.

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

Prerequisites

While the use of Amazon Connect contact center is not required for using Unified Profiles for Travelers and Guests on AWS, some features of the solution (setting up deterministic matching rules and feed traveler profiles to Amazon S3) must be set up using the Amazon Connect Customer Profiles console. This console is only accessible by creating an Amazon Connect instance and mapping your created domain to it. To create an instance if you do not have one already, follow Step 1: Set identity in the *Amazon Connect Administrator Guide*.

> ⓘ **Note**
>
> You will not be charged for creating an Amazon Connect instance.

Step 1: 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 15 minutes

1. Sign in to the AWS Management Console and select the button to launch the `ucp.template` AWS 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. For information about naming character limitations, see IAM and STS Limits in the *AWS Identity and Access Management User Guide*.

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

Parameter	Default	Description
eventbridgeActivated	`true`	Configure the EventBridge integration. If set to true, all profile changes will generate events on a dedicated Amazon EventBridge bus created by the solution.
industryConnectorB ucketName	None	If provided, the solution will set up data replicati on between the Travel and Hospitality Connector Catalog on AWS solution and this solution. See the Integrating with the Travel and Hospitality Application Connectors Catalog on AWS section for more details.
partitionStartDate	2023/08/01	If you load historical data into the Unified Profiles for Travelers and Guests on AWS, provide the date of the earliest datapoint to allow data to be processed.
errorTTL	7	The number of days to retain ingestion error records.

Parameter	Default	Description
skipJobRun	`true`	If set to `false`, the jobs will run incrementally every two hours, Job-based ingestion is recommended primarily for historical data ingestion . By default, the details flag is set to `true`. User can start these jobs from the solution's frontend screen.
inputStreamMode	`ON_DEMAND`	The input stream capacity mode. If you plan to ingest a large amount of data (10s of millions) through the real-time feed, change this value to `PROVISIONED` .
inputStreamShards	`100`	Default shard count for the input stream (ignored if capacity mode is `ON_DEMAND`).
ingestorShardsCount	`10`	Default shard count for the ingestor stream. This stream is set to `PROVISIONED` to throttle the traffic going into Amazon Connect Customer Profiles. Validate this value with your account team if above 10.

Parameter	Default	Description
exportStreamMode	`ON_DEMAND`	The export stream capacity mode. If you plan to ingest a large amount of data (10s of millions) through the real-time feed, change this value to `PROVISIONED` .
exportStreamShards	`10`	Default shard count for the export stream (ignored if capacity mode is `ON_DEMAND`).

6. Choose **Next**.

7. On the **Configure stack options** page, chose **Next**.

8. On the **Review and create** page, review and confirm the settings. Check the box acknowledging that the template will create 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 15 minutes.

> ⓘ **Note**
>
> In addition to the primary Lambda functions, this solution includes the `solution-helper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.
> When you run this solution, you will notice both Lambda functions in the AWS console. Only the core functions are regularly active; however, you must not delete the `solution-helper` function, as it is necessary to manage associated resources.

Step 2: Create users

Go to the Amazon Cognito console with an IAM-user; that has permissions to create users and groups and locate the userpool created by the solution (it should be prefixed by `ucp-`). Create an Amazon Cognito user in this userpool for your administrator.

1. Navigate to the [Amazon Cognito console](#) with an IAM user that has permissions to create users and groups.

2. Locate the userpool created by the solution (prefixed by `ucp-`).

3. Create an Amazon Cognito user in this userpool for your administrator.

Additionally, you can create other users in this userpool to give access to the solution.

> ⓘ **Note**
>
> By default, users will be able to login but will not see any traveler data. (Refer to the Amazon Cognito section for steps on how to give user scoped permission to see traveler data.)

Step 3: Create an Amazon Connect Customer Profiles domain

1. Sign in to the solution frontend (find the URL in the **Outputs** tab of the CloudFormation deployment) with the created user.

2. On the **Stacks** page, select the stack.

3. Select **New Domain** from the list.

4. Enter a domain name (must follow the snake case convention). Take note of the domain name which will be required for multiple steps.
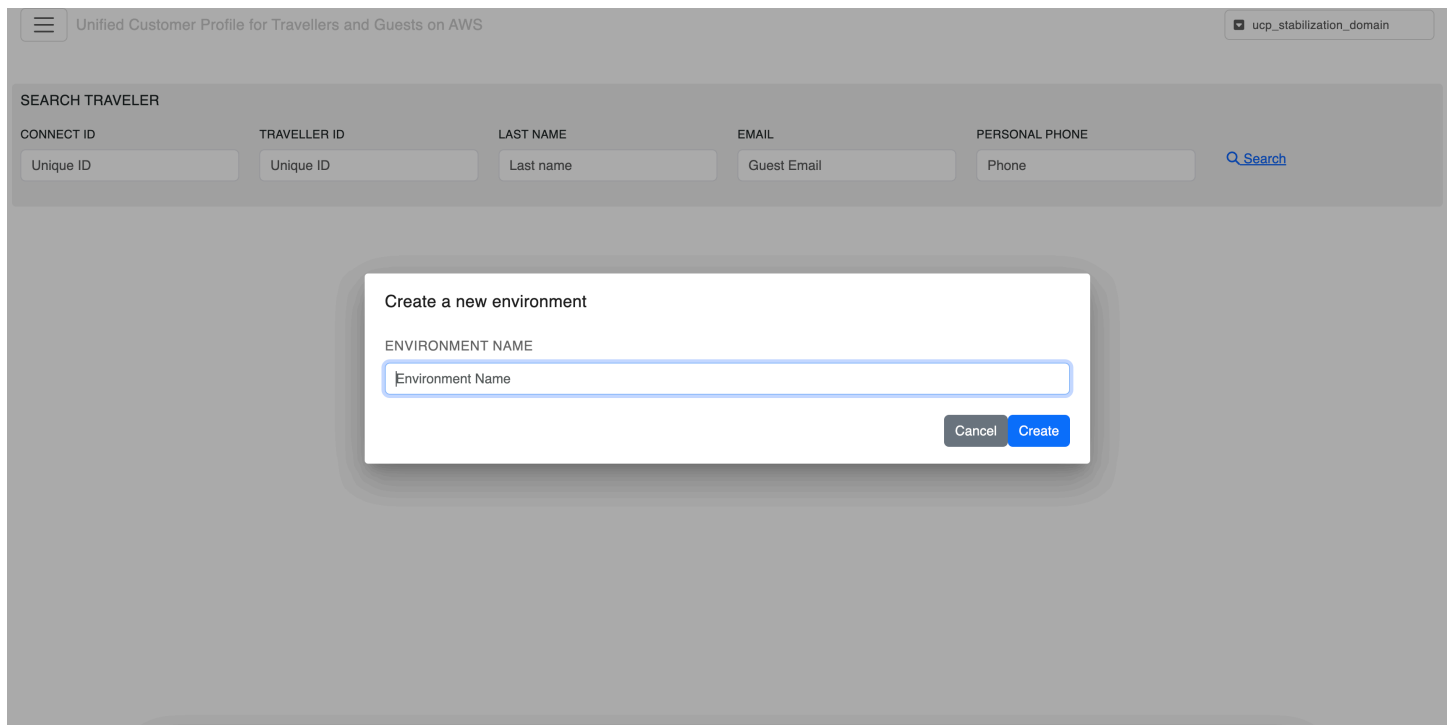
Step 4: Provide user permissions

1. Navigate to the Amazon Cognito console.

2. Locate the permission group prefixed by your domain name. Any user added to this group will have unrestricted access to the traveler data.

3. You can create additional groups with more narrow permissions by following the instructions in the Amazon Cognito section.

Step 5: Map the Customer Profiles domain to your Amazon Connect instance

1. Navigate to the Amazon Connect console.

2. Go to Customer Profiles and associate your Amazon Connect Customer Profiles domain with the instance you created.

   If you do not have an Amazon Connect instance, create one, then select the instance, and go to the Customer Profiles section.

   At the Customer Profiles section you should not be associated with any domains.
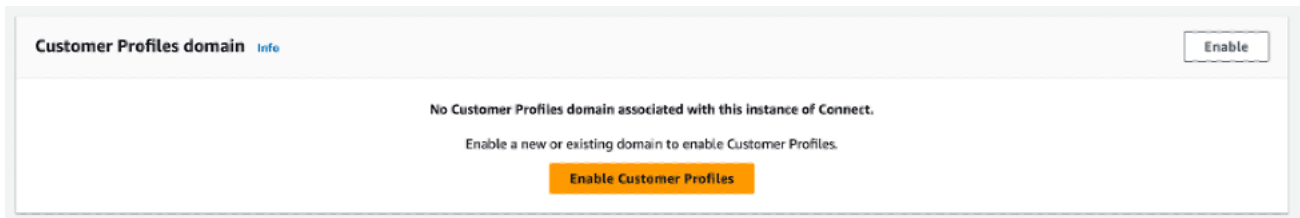
Customer Profiles - no domain associated

> ⚠ **Important**
>
> If you are not attached to your intended domain (the domain created in the section
> called "Step 3: Create an Amazon Connect Customer Profiles domain") detach from the
> existing domain.

3. Choose Enable, select Use existing domain and select the domain from the list to attach the
   domain from the section called "Step 3: Create an Amazon Connect Customer Profiles domain".
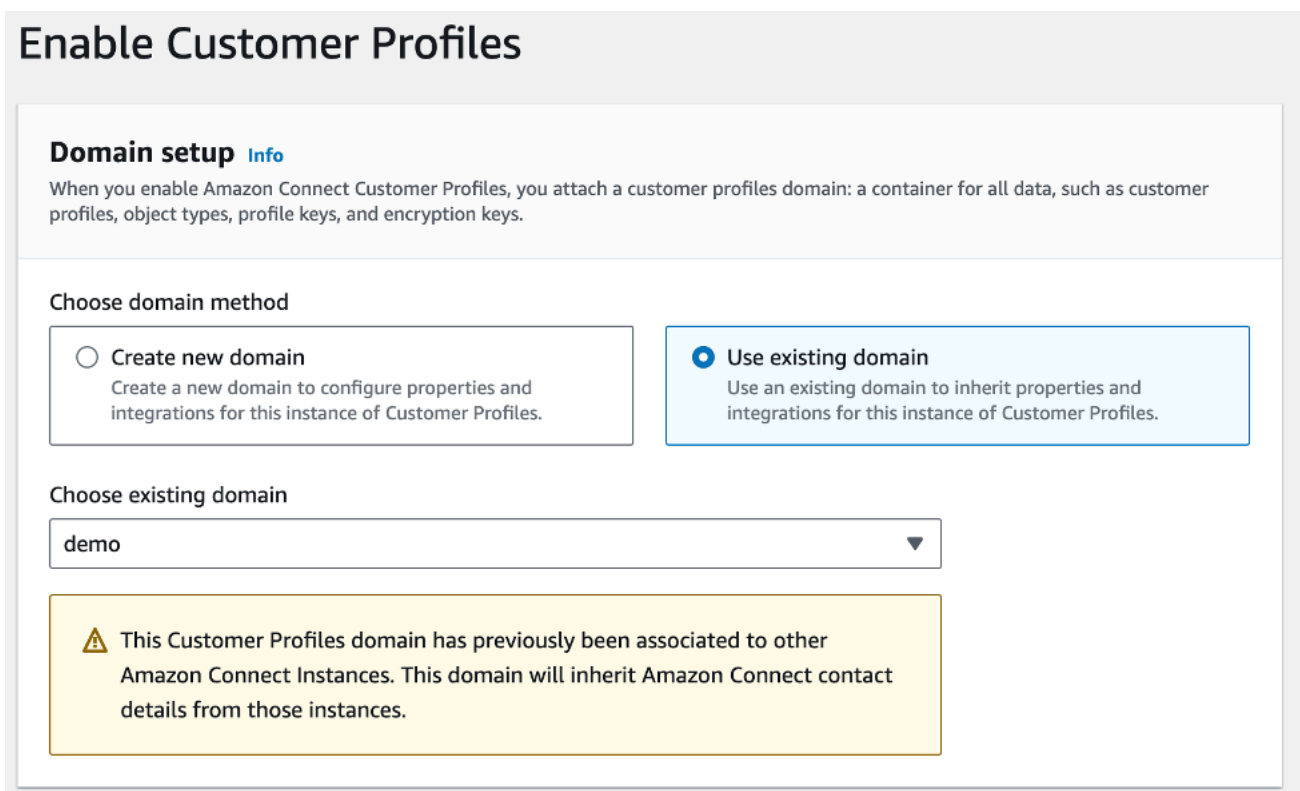
4. Leave the remaining options as set by the solution and choose **Enable Customer Profiles**. This
   allows you to configure identity resolution rules.

Step 6: Create match and merge rules

To create match and merge rules for your domain, follow the steps to Set up rule-based matching for Identity Resolution in the *Amazon Connect Administrator Guide.* All domains created with the solution have AI-based identity resolution activated. You can deactivate this functionality if you do not need it and would like to lower the cost of the solution.

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.
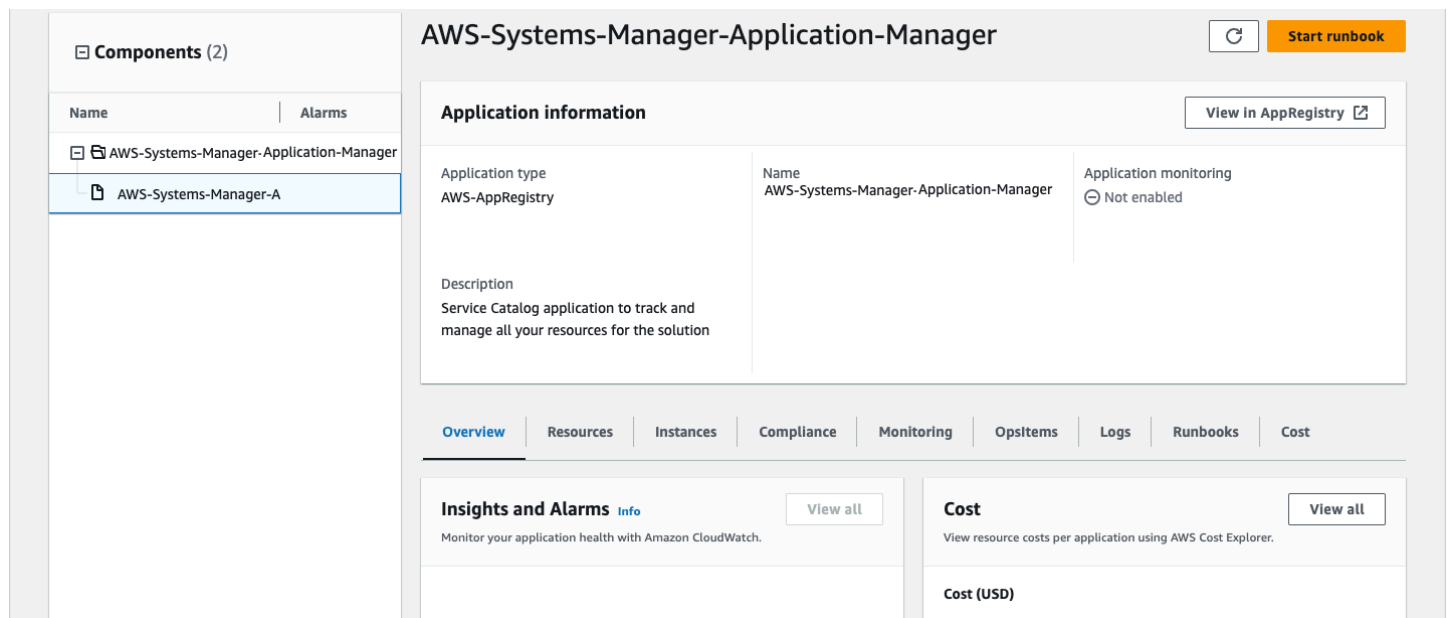
Solution stack in Application Manager

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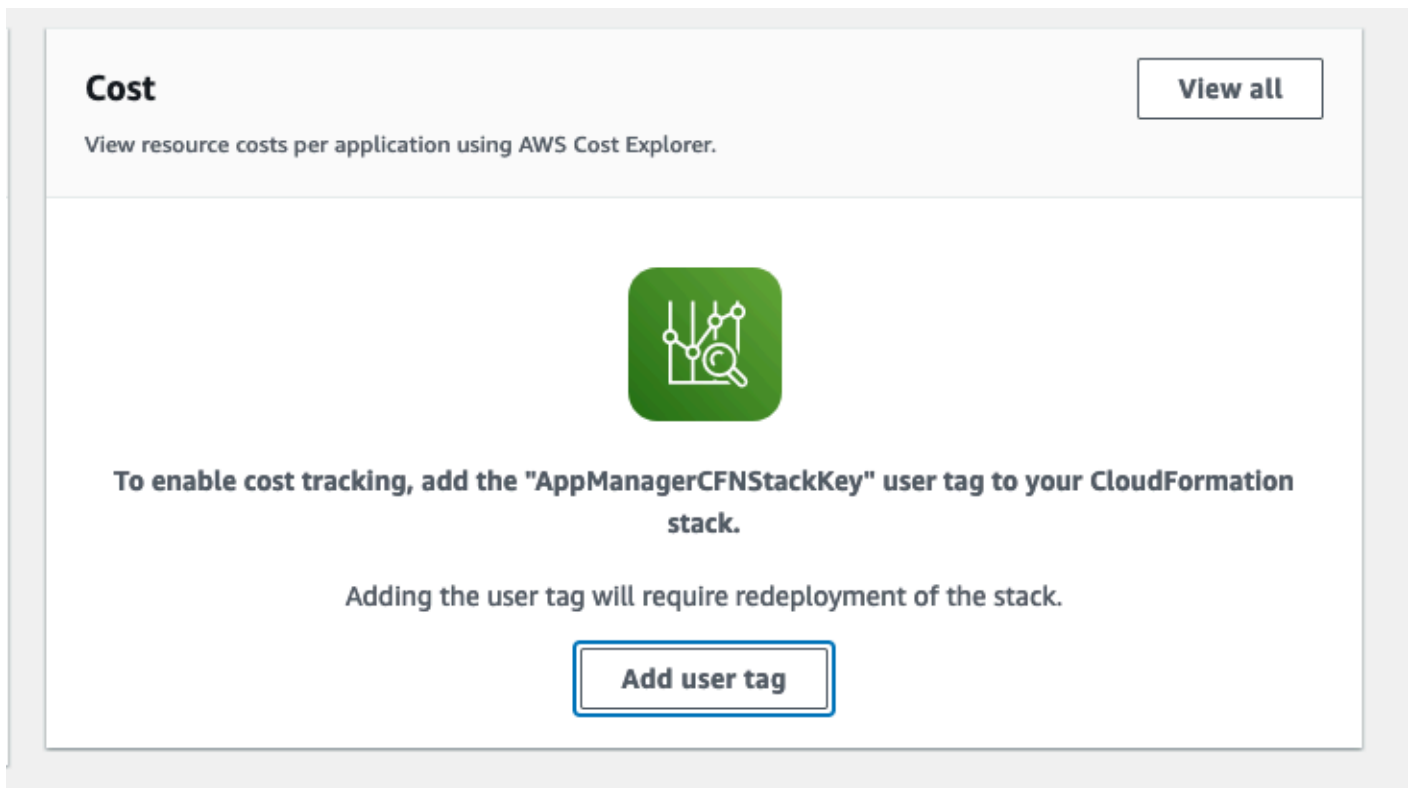
5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

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 solution's CloudFormation stack to get the latest version of the solution's framework.

1. Sign in to the AWS CloudFormation console, select your existing Unified Profiles for Travelers and Guests on AWS CloudFormation stack, and select **Update**.

2. Select **Replace current template**.

3. Under **Specify template**:

   a. Select **Amazon S3 URL**.

   b. Copy the link of the `ucp.template` the section called "AWS CloudFormation 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**. Choose **Next** again.

4. Under **Parameters**, review the parameters for the template and modify them as necessary. For details about the parameters, see Launch the stack.

5. Choose **Next**.

6. On the **Configure stack options** page, choose **Next**.

7. On the **Review** page, review and confirm the settings. Select the box acknowledging that the template will 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 UPDATE_COMPLETE status in approximately 15 minutes.

Troubleshooting

This section provides troubleshooting instructions for deploying and using the solution.

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

Problem: Ingestion errors

All ingestion errors are ingested into a DynamoDB database and displayed in real time on the **Settings** screen.

Common errors and resolution

Unknown object type

This error occurs when the name of the object type is invalid. For the supported list of object types, see Kinesis wrapper schema.

DomainName not found

This error occurs when the name of the Amazon Connect Customer Profiles domain cannot be found in the account. Verify that the DomainName is correct and that the domain has been created. For details, see [Step 3: Create an Amazon Connect Customer Profiles domain](#).

Transformer_exception 'stringValue'

The error message contains StringValue for a Clickstream object

This occurs when a Clickstream attribute has a "string" type and the StringValue field is not populated. Validate that your Clickstream attribute type and fields are consistent. For details, see [Web and mobile events](#).

'NoneType' object is not iterable

This error occurs when the solution ingest expects an array type and receives a null type. Verify that no field has been set to null in the input record. For the supported list of object types, see [Kinesis wrapper schema](#).

Unconverted data remains: Z

This error occurs when the solution ingests a timestamp that does not match the expected ISO 8601 format (including milliseconds).

Incorrect:

```
{
  "modelVersion": "1.0",
  "lastUpdated": "0001-01-01T00:00:00Z",
  ...
}
```

Correct:

```
{
  "modelVersion": "1.0",
  "lastUpdated": "0001-01-01T00:00:00.000Z",
  ...
}
```

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 Unified Profiles for Travelers and Guests on AWS solution from the AWS Management Console or by using the AWS Command Line Interface. AWS Solutions Implementations do not automatically delete Amazon S3 buckets in case you have stored data to retain.

Prerequisites

Prior to deleting the stack. Make sure to complete the following:

- Delete any rules associated to the EventBridge bus.

- Remove Amazon Cognito user pool deletion protection.

- Empty all S3 buckets (buckets with data will prevent stack deletion).

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, 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>* S3 buckets.
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
```

Use the solution

This section outlines the key concepts and functions for using the solution.

Sending data to the real-time stream

The preferred way to send traveler data to Unified Profiles for Travelers and Guests on AWS is to use the real-time ingestion flow. It allows the solution to ingest, resolve, and activate travelers within seconds of the traveler journey event. The solution supports receiving seven business objects that must follow the AWS schema provided with the solution.

> **ⓘ Note**
>
> The *required* property is added by default by our schema generator but none of the fields are required. The design principle is *send what you have and the solution will do its best to create a profile or append to an existing profile*. The cleanup process happens downstream.

Air booking

A representation of an airline booking as generated by PSS providers like Amadeus, Altea, or Sabre Sonic. The object contains a list of segments representing flight legs. Each of these segments will be ingested as an Amazon Connect Customer Profiles (ACCP) custom object. The schema for can be found in air_booking.schema.json.

Passenger profile

A representation of the passenger profile containing the passenger PIIs and loyalty information. The traveler object has an ID field that we use to create the traveler ID. If you have any ID from the source system that could help identify the guest, provide it in the ID field as it will make the ID resolution much more efficient. The schema for passenger profiles can be found in pax_profile.schema.json.

Hotel booking

A representation of a hotel guest booking record as generated by mainstream central reservation systems like Amadeus or Sabre. Each hotel booking object must have at least one segment. The hotel booking object is split into product/guest records where each record must have a unique ID.

The ID is built form the combination of the hotel code, the booking confirmation number (ID) ,the segment ID, the segment start date, and the product ID. These parameters must be carefully chosen from the source system in order to ensure data accuracy. The schema for hotel bookings can be found in hotel_booking.schema.json.

Hotel guest

A representation of the guest profile containing the guest PIIs and loyalty information (profile and associated transactions). The guest object has an ID field that we use to create the traveler ID. If you have any ID from the source system that could help identify the guest, provide it in the ID field as it will make the ID resolution much more efficient. The schema for hotel guests can found in guest_profile.schema.json.

Hotel stay

A representation of some of the hotel property management system folio content. Mainly used (in the context of this solution) to convey in-house purchases made by the guest.

The solution uses the following constraints:

- Each item under the revenue field must have a unique `"type"` field.
- Each `hotel-stay` object must have at least one revenue item.
- We recommend populating the startDate field with the hotel stay check in date.
- If the GuestId provided matches an existing guest and PII (names, email, Phones…) aren't provided with the object, the PII will be deleted from the profile. For this reason, we recommend using a unique guest ID and auto merging rules to aggregate `hotel-stay` objects.

The schema for hotel stays can be found in hotel_stay_revenue.schema.json.

Example object:

```
{
  "domain":"ucp_stabilization_domain",
    "data":
    {
    "guestId":"s4qy8dn8ahjm3wz94ht7",
    "startDate":"2023-8-8",
    "revenue":[
    {
    "type":"food and beverage: Chicken Green Goddess Salad",
```

```
        "description":"Chicken Green Goddess Salad",
        "currency":{
        "code":"USD"
        },
        "amount":9.25,
        "date":"2023-08-08T14:39:24.395Z"
        },
        {
        "type":"food and beverage: Big Don® Italian",
        "description":"Big Don® Italian",
        "currency":
        {
        "code":"USD"
        },
        "amount":13.25,
        "date":"2023-08-08T14:39:24.395Z"
                }
            ]
        },
        "objectType":"hotel_stay"
    }
```

Web and mobile events

A representation of events coming from web and mobile channels. The table has the event names and attributes supported by the solution.

Event names	Event descriptions
start_session	The traveler initiates a new session by accessing the travel booking site or mobile app. A session ID is created.
login	The traveler ID for signin.
logout	The traveler signs out.
search_destination	The traveler searches for a destination (could be a county, a city, an airport, or a specific hotel property). Typically, a geolocation service.

Event names	Event descriptions
search_experience	The traveler searches for a travel experience (could be a cruise, a tour, an adventure vacation…). This search is not specific to a destination.
select_destination	The traveler selects one of the destinations returned by the search service used.
select_experience	The traveler selects one of the experiences returned by the search service used.
select_origin	The traveler selects an origin for the trip (departure airport, car rental pick-up location…).
select_n_traveller	The traveler selects the number of travelers on the trip.
select_start_date	The traveler selects a start date for the trip (check-in date for a hotel, departure date for a flight, pick-up date for a car rental, or a cruise).
select_end_date	The traveler selects an end date for the trip (return date for a flight, a car rental, checkout date for a hotel…).
select_start_time	Optional time to be used for hotel, alternate accommodation check-in, or car rental pick-up.
select_end_time	Optional time to be used for hotel, alternate accommodation checkout, or car rental return.
select_n_rooms	The traveler selects a number of rooms for a stay or cabins for a cruise.

Event names	Event descriptions
select_n_nights	The traveler selects a number of nights for a hotel stay or a cruise.
search_flight	Search for a flight.
search_multi_availability	Search for availability for multiple properties.
search_single_availability	Search for availability in one property.
view_product	The traveler views a product.
select_product	The traveler selects a product from the search results.
start_booking	The traveler starts a booking.
update_booking	The traveler updates a booking.
confirm_booking	The traveler confirms a booking.
cancel_booking	The traveler cancels a booking.
retrieve_booking	Retrieve a booking.
search_booking	Search for a booking.
ignore_booking	Ignore a booking.
regret	Regret event.
search_add_on	Search for an add-on to add to the reservation.
view_add_on	View an add-on.
select_add_on	Select an add-on from the search results.
remove_add_on	Remove an add-on from the reservation.

Event names	Event descriptions
custom	Custom event.
cross_geofence	Cross geofence event.
EVENT ATTRIBUTES LODGING	**EVENT ATTRIBUTES DESCRIPTION**
room_type	Room Type Code "DBL", "KING"
rate_plan	Rate Plan Code "BAR"
checkin_date	Check-in date
checkout_date	Check-out date
num_nights	Length of stay
hotel_code	Hotel identifier
hotel_code_list	List of hotels identified (for multi-availability)
hotel_name	Human-readable hotel name
products	Products ["Fare", "Fare", "BGB1", "CRRB", "BGBN", "BSNS", "BGB1", "CRRB", "BGBN", "BSNS", "SEAT", "SEAT", "MCES"] Array
products_prices	Products Price ["Room-Rate", "Room-Rate", "BKFST", "CMPGNE"] Array
quantities	Product Quantity ["3", "3", "3"] Array
destination	Destination searched by the guest
num_guest_adults	Number of Adults "2"
num_guest_children	Number of Children "1"
num_guests	Number of Passengers "3"

Event names	Event descriptions
EVENT ATTRIBUTES AIR	**EVENT ATTRIBUTES DESCRIPTION**
`fare_class`	Fare Class "A"
`fare_type`	Fare Type "P"
`flight_market`	Flight Market "DOM,DOM"
`flight_segments_departure_date_time`	Flight Segment Date Time ["2022-09-01 13:36:00", "2022-09-03 13:31:00"] Array
`flight_segments_arrival_date_time`	Flight Segment Date Time ["2022-09-01 13:36:00", "2022-09-03 13:31:00"] Array
`flight_segment_sku`	Flight Segment Number ["Y4 748", "Y4 107"] Array
`flight_segments`	Flight Segments ["MEX-ACA", "ACA-MEX"] Array
`flight_type`	Flight Type "Round Trip"
`origin_date`	Origin Date "2022-09-07"
`origin_date_time`	Origin Flight Date Time "2022-09-01 13:36:00"
`origin_flight_route`	Flight Origin "QMX-AGU"
`returning_date`	Return Flight Date "2022-09-03"
`returning_date_time`	Return Flight Date Time "2022-09-03 13:31:00"
`returning_flight_route`	Flight Return "AGU-QMX"
`products`	Products ["Fare", "Fare", "BGB1", "CRRB", "BGBN", "BSNS", "BGB1", "CRRB", "BGBN", "BSNS", "SEAT", "SEAT", "MCES"] Array

Event names	Event descriptions
products_prices	Products Price ["Room-Rate", "Room-Rate", "BKFST", "CMPGNE"] Array
quantities	Product Quantity ["3", "3", "3"] Array
num_pax_adults	Number of Adults "2"
num_pax_inf	Number of Infants "1"
num_pax_children	Number of Children "1"
pax_type	Passenger Type "passenger"
total_passengers	Number of Passengers "3"
EVENT ATTRIBUTES COMMON	**EVENT ATTRIBUTES DESCRIPTION**
customer_birthdate	Contains the customer's date of birth. "1981-03-03T00:00:00"
customer_country	Contains the customer's country. "MX"
customer_email	Contains the customer's email address. "teal@test.com"
customer_first_name	Contains the customer's first name. "John"
customer_gender	Contains the customer's gender. "Male"
customer_id	Contains the customer's ID. 123
customer_last_name	Contains the customer's last name. "Doe"
customer_nationality	Contains the customer's nationality. "MX"
customer_phone	Contains the customer's phone number. "527523695215"
customer_type	Contains the customer's type. "Anonymous"

Event names	Event descriptions
language_code	Contains the language code. "en-US"
loyalty_id	Contains the loyalty program identifier
currency	Contains the currency. "MXN"
ecommerce_action	Contains the ecommerce action. "Booking Flow"
order_payment_type	Payment Type "Visa;VI,Visa;VI,Visa;VI"
order_promo_code	Promo Code "VOLH50"
page_name	Page Name "FlightSearch"
page_type_environment	Page Type test
transaction_id	ID of the purchase transaction
booking_id	Airline PNR (ex X7Z1TI) or Hotel Booking ID
url	Value of the page URL. volaris.com
geofence_latitude	Contains the geofence latitude. "19.4"
geofence_longitude	Contains the geofence longitude. "99.4"
geofence_id	Contains the geofence id. "1"
geofence_name	Contains the geofence name. "1"
poi_id	Place ID in geolocation API.
customer_event_name	If event name different than tealium_event.
custom	Reserved mapping for custom attribute names.
EVENT ATTRIBUTES TYPES	**EVENT ATTRIBUTES DESCRIPTION**

Event names	Event descriptions
string	Use this value to specify that this attribute is a single string value.
strings	Use this value to specify that this attribute is an array of strings.
number	Use this value to specify that this attribute is a single number value.
numbers	Use this value to specify that this attribute is an array of numbers.

Constraints:

- For events to be aggregated in the same profile, they must have the same sessionId field.

The schema for click events can be found in clickevent.schema.json.

Customer service interaction

A representation of an interaction between the traveler and your customer service organization. This object can hold interactions with multiple channels (SMS, email, chat, and voice transcript).

Thes schema for customer service interactions can be found in customer_service_interaction.schema.json.

Kinesis wrapper schema

To send these objects to the solution, they must be wrapped in a JSON envelope with the following top-level keys:

- **objectType** - Business object name within hotel_booking, pax_profile, air_booking, guest_profile, hotel_stay, and clickstream.
- **modelVersion** - Current model version. Use 1.0.
- **data** - The business object following the schema above.
- **domain** - The name of the Amazon Connect Customer Profiles domain created.

- **mode** - If set to `partial`, the solution seeks to override an existing Amazon Connect Customer Profiles object with some of the data provided in the data field. If the business object contains an ACCP object in the domain with matching object IDs and profile IDs, that would be the object that is to be overridden with fields set in `partialModeOptions`. See that for more information. If set to `merge`, the solution will merge the incoming profile with the provided `MergeModeProfileID`.

- **timestamp** - Provides a timestamp to allow replayability of the traffic (format: `"%Y-%m-%dT%H:%M:%S.%fZ"`).

- **uid** - Provide a unique ID to allow replayability of the traffic.

- **partialModeOptions** - Contains the `fields` property, which is an array of fields to override. It follows the following format: `"accp_object_name"."property"` (note, the `accp_object_name` is not the business object name). For example, to override the points and level properties of the loyalty program for the guest profile, the fields property would be: `"hotel_loyalty.level","hotel_loyalty.points"`. You can also set the property to wildcard (`"*"`) to override all the fields in the ACCP object.

Example wrapper message: (For readability, the business object data is omitted.)

```
{
 "objectType": "air_booking",
 "modelVersion": "1.0",
 "data": {…},
 "domain": "my_domain_name",
 "mode": "",
 "timestamp": "2022-08-22T09:08:00.123456",
 "uid": "d2bd8928-23da-4240-bfbc-ee03e863cf51"
}
```

Example wrapper message in partial mode:

```
{
 "objectType": "air_booking",
 "modelVersion": "1.0",
 "data": {…},
 "domain": "my_domain_name",
 "mode": "partial",
 "timestamp": "2022-08-22T09:08:00.123456",
 "uid": "d2bd8928-23da-4240-bfbc-ee03e863cf51",
 "partialModeOptions": {
```

```
      "fields": ["air_loyalty.points"]
   }
}
```

ACCP objects and their properties

The data used to override the object is present in the data field. The solution will find the matching object and profile IDs to do the update.

Object name	Fields
air_booking	accp_object_id, first_name, nationality_code, email_type, last_updated_by, address_billing_line3, address_line4, price, address_billing_postal_code, traveller_id, last_updated, address_mailing_country, company, address_mailing_line3, address_line1, address_business_country,

Object name	Fields
	address_business_postal_code,
	address_mailing_postal_code,
	address_mailing_state_province,
	address_state_province,
	arrival_date,
	address_business_line3,
	cc_name,
	address_billing_state_province,
	from,
	address_line3,
	to,
	segment_id,
	job_title,
	honorific,
	address_billing_line2,
	pss_id,
	address_postal_code,
	cc_exp,
	phone_mobile,
	booking_id,
	address_business_state_province,

Object name	Fields
	payment_type,
	address_mailing_line2,
	phone_business,
	phone_home,
	cc_token,
	address_mailing_city,
	email_business,
	language_name,
	address_city,
	address_billing_line1,
	address_billing_line4,
	middle_name,
	gender,
	departure_time,
	address_business_line2,
	pronoun,
	address_mailing_line1,
	address_mailing_line4,
	address_line2,
	status,
	gds_id,

Object name	Fields
	last_name,
	email,
	cc_cvv,
	language_code,
	object_type,
	arrival_time,
	model_version,
	channel,
	address_billing_city,
	nationality_name,
	address_type,
	address_billing_country,
	address_business_line1,
	departure_date,
	address_business_line4,
	date_of_birth,
	address_country,
	phone,
	address_business_city,
	flight_number,
	cc_type,

Object name	Fields
	phone_type, last_update_channel_id, creation_channel_id, last_booking_id,
air_loyalty	accp_object_id, last_updated_by, traveller_id, level, last_updated, program_name, miles, miles_to_next_level, joined, object_type, model_version, id,

Object name	Fields
clickstream	object_type,
	aws_account_id,
	model_version,
	traveller_id,
	accp_object_id,
	last_updated,
	session_id,
	event_timestamp,
	event_type,
	event_version,
	arrival_timestamp,
	user_agent,
	custom_event_name,
	customer_birthdate,
	customer_country,
	customer_email,
	customer_first_name,
	customer_gender,
	customer_loyalty_id,
	customer_last_name,
	customer_nationality,

Object name	Fields
	customer_phone,
	customer_type,
	currency,
	products,
	products_prices,
	quantities,
	ecommerce_action,
	order_payment_type,
	order_promo_code,
	page_name,
	page_type_environment,
	transaction_id,
	url,
	fare_class,
	fare_type,
	flight_segments_departure_date_time,
	flight_numbers,
	flight_market,
	flight_type,
	origin_date,

Object name	Fields
	origin_date_time,
	return_date,
	return_date_time,
	return_flight_route,
	num_pax_adults,
	num_pax_inf,
	num_pax_children,
	pax_type,
	total_passengers,
	room_type,
	rate_plan,
	checkin_date,
	checkout_date,
	num_nights,
	num_guests,
	hotel_code,
	hotel_code_list,
	hotel_name,
	destination,
	num_guest_adults,
	num_guest_children,

Object name	Fields
	flight_segments_arrival_date_time,
	flight_segment_sku,
	flight_segments,
	origin_flight_route,
	returning_date,
	returning_date_time,
	returning_flight_route,
	customer_id,
	language_code,
	loyalty_id,
	booking_id,
	geofence_latitude,
	geofence_longitude,
	geofence_id,
	geofence_name,
	poi_id,
	customer_event_name,
	custom,

Object name	Fields
customer_service_interaction	model_version, object_type, last_updated, traveller_id, accp_object_id, channel, loyalty_id, first_name, last_name, email, phone_number, interaction_type, start_time, end_time, duration, status, language_code, language_name, conversation,

Object name	Fields
email_history	accp_object_id,
	last_updated_by,
	traveller_id,
	last_updated,
	type,
	address,
	object_type,
	model_version,

Object name	Fields
guest_profile	accp_object_id,
	nationality_code,
	first_name,
	email_type,
	last_updated_by,
	address_billing_line3,
	address_line4,
	address_billing_postal_code,
	traveller_id,
	last_updated,
	address_mailing_country,
	company,
	address_mailing_line3,
	address_business_country,
	address_line1,
	address_business_postal_code,
	address_mailing_postal_code,
	address_mailing_state_province,
	created_on,
	address_state_province,
	address_business_line3,

Object name	Fields
	address_billing_state_province,
	created_by,
	address_line3,
	job_title,
	honorific,
	address_billing_line2,
	address_postal_code,
	phone_mobile,
	address_business_state_province,
	address_mailing_line2,
	phone_home,
	phone_business,
	address_mailing_city,
	email_business,
	language_name,
	address_city,
	address_billing_line1,
	address_billing_line4,
	middle_name,
	gender,
	address_business_line2,

Object name	Fields
	address_mailing_line1,
	pronoun,
	address_mailing_line4,
	address_line2,
	last_name,
	language_code,
	email,
	object_type,
	model_version,
	nationality_name,
	address_billing_city,
	address_type,
	address_billing_country,
	address_business_line1,
	phone_type,
	address_business_line4,
	date_of_birth,
	address_country,
	phone,
	address_business_city,

Object name	Fields
hotel_booking	accp_object_id, first_name, nationality_code, totalAfterTax, email_type, address_line4, last_updated_by, address_billing_line3, n_nights, product_id, address_billing_postal_code, traveller_id, last_updated, address_mailing_country, n_guests, status, company, address_mailing_line3, address_line1, address_business_country, address_business_postal_code,

Object name	Fields
	address_mailing_postal_code,
	address_mailing_state_province,
	address_state_province,
	address_business_line3,
	cc_name,
	address_billing_state_province,
	room_type_description,
	address_line3,
	job_title,
	honorific,
	address_billing_line2,
	address_postal_code,
	room_type_name,
	cc_exp,
	phone_mobile,
	booking_id,
	address_business_state_province,
	payment_type,
	totalBeforeTax,
	attribute_names,
	address_mailing_line2,

Object name	Fields
	phone_business, phone_home, cc_token, address_mailing_city, email_business, language_name, address_city, address_billing_line1, attribute_codes, address_billing_line4, middle_name, gender, attribute_descriptions, room_type_code, address_business_line2, pronoun, address_mailing_line1, check_in_date, address_mailing_line4, address_line2, gds_id,

Object name	Fields
	last_name,
	email,
	cc_cvv,
	language_code,
	object_type,
	hotel_code,
	model_version,
	address_billing_city,
	nationality_name,
	crs_id,
	address_type,
	address_billing_country,
	pms_id,
	address_business_line1,
	address_business_line4,
	date_of_birth,
	address_country,
	phone,
	address_business_city,
	cc_type,
	phone_type,

Object name	Fields
	creation_channel_id, last_update_channel_id, last_booking_id,
hotel_loyalty	accp_object_id, last_updated_by, last_updated, level, traveller_id, program_name, points, points_to_next_level, units, joined, model_version, object_type, id,

Object name	Fields
hotel_stay	accp_object_id,
	first_name,
	last_updated_by,
	date,
	last_updated,
	traveller_id,
	type,
	created_on,
	created_by,
	amount,
	currency_symbol,
	booking_id,
	description,
	currency_name,
	currency_code,
	email,
	last_name,
	model_version,
	object_type,
	hotel_code,
	start_date,

Object name	Fields
	id, phone,

Object name	Fields
loyalty_transaction	object_type,
	model_version,
	last_updated,
	accp_object_id,
	traveller_id,
	last_updated_by,
	category,
	points_offset,
	points_unit,
	origin_points_offset,
	qualifying_point_offset,
	source,
	booking_date,
	order_number,
	product_id,
	expire_in_days,
	amount,
	amount_type,
	voucher_quantity,
	corporate_reference_number,
	promotions,

Object name	Fields
	location, activity_day, to_loyalty_id, from_loyalty_id, organization_code, event_name, document_number, corporate_id, program_name,

Object name	Fields
pax_profile	accp_object_id,
	nationality_code,
	first_name,
	email_type,
	last_updated_by,
	address_billing_line3,
	address_line4,
	address_billing_postal_code,
	traveller_id,
	last_updated,
	address_mailing_country,
	company,
	address_mailing_line3,
	address_business_country,
	address_line1,
	address_business_postal_code,
	address_mailing_postal_code,
	address_mailing_state_province,
	created_on,
	address_state_province,
	address_business_line3,

Object name	Fields
	address_billing_state_province,
	created_by,
	address_line3,
	job_title,
	honorific,
	address_billing_line2,
	address_postal_code,
	phone_mobile,
	address_business_state_province,
	address_mailing_line2,
	phone_home,
	phone_business,
	address_mailing_city,
	email_business,
	language_name,
	address_city,
	address_billing_line1,
	address_billing_line4,
	middle_name,
	gender,
	address_business_line2,

Object name	Fields
	address_mailing_line1,
	pronoun,
	address_mailing_line4,
	address_line2,
	last_name,
	language_code,
	email,
	object_type,
	model_version,
	nationality_name,
	address_billing_city,
	address_billing_country,
	address_business_line1,
	phone_type,
	address_business_line4,
	date_of_birth,
	address_country,
	phone,
	id,
	address_business_city,
	address_type,

Object name	Fields
phone_history	accp_object_id, last_updated_by, country_code, traveller_id, last_updated, type, number, object_type, model_version,

Fields corresponding to the `accp_object_id` and the `traveller_id` will not be overwritten, since those fields identify the object to be updated.

- **mergeModeProfileID** - String that contains the profile ID to merge with when the mode is set to `merge`.

To send the data to the solution, use the solution's Kinesis data stream provided in the CloudFormation output under the **kinesisStreamNameRealTime** key. Then use this tutorial to send data to the data stream.

> **ⓘ Note**
>
> All objects sent to the solution using real-time Kinesis ingestion are backed up in Amazon S3 under a `"{domain}/"` path. This allows the solution to replay the traffic in case of errors downstream that cause loss of data.

General troubleshooting
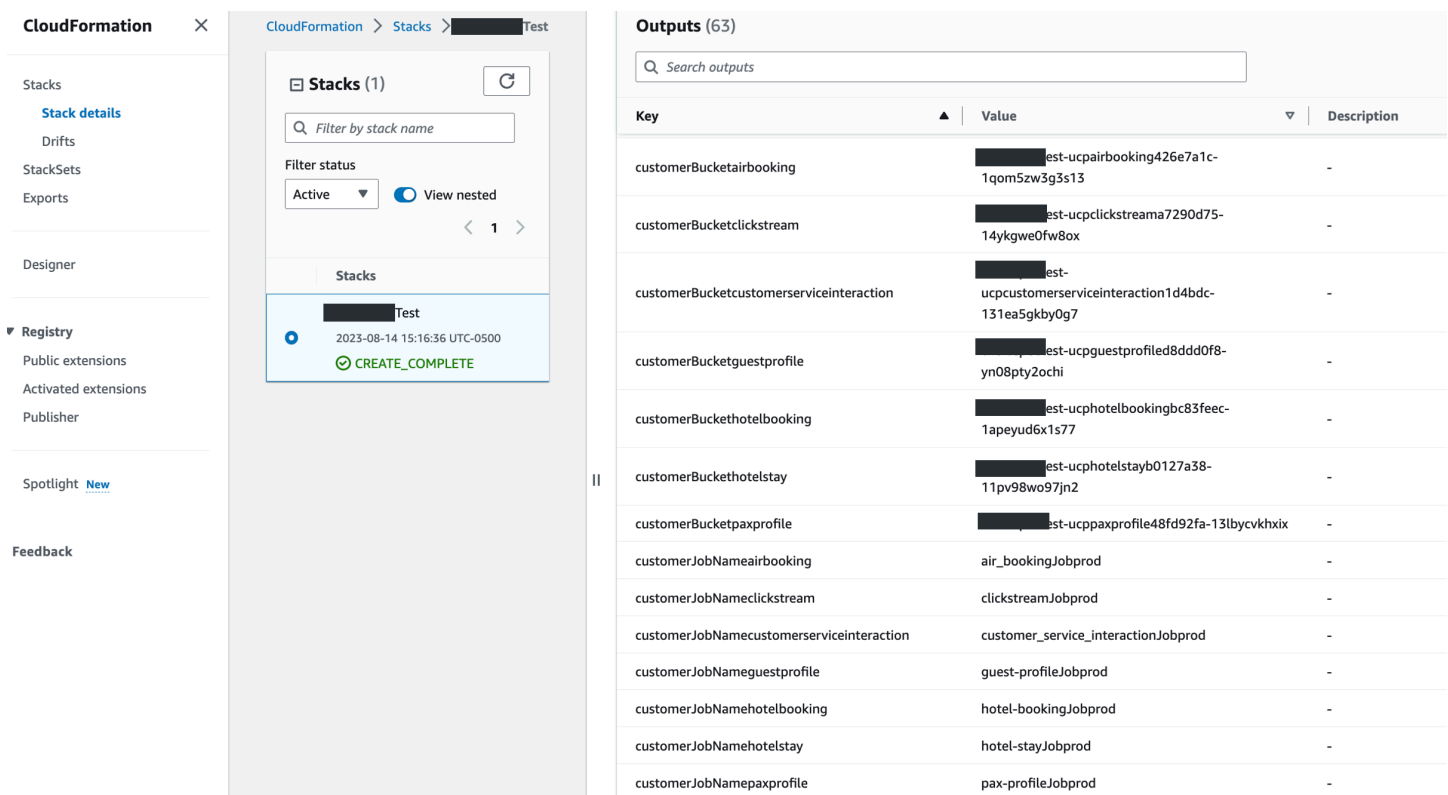
Avoid null types. (For example, emptylist, strings or empty objects are preferred to null types which will rate transformation issues.) Using null types in input business objects will lead to the NoneType object is not iterable error viewable on the settings screen.

Sending data to S3 for batch ingestion

For data ingested in batch, you can upload the business object (following the JSON schema provided in the Sending data to the real-time stream) in a dedicated Amazon S3 bucket created by the solution. The upload must meet the following constraints:

- The root folder must be the ACCP domain name

- The path must contain the lastUpdate timestamp provided in the following format: YYYY/MM/DD

- Objects must be ingested as JSONL (line-break-separated JSON objects where JSON data must not be prettified)

The name of the Amazon S3 bucket to upload the data can be found in the CloudFormation **customerBucket*<businessObjectName>*** output. For example, the `customerBucketair_booking` output key contains the bucket name of where to upload the air booking objects.

Example of the CloudFormation output

Buckets can also be opened directly from the solution frontend **Settings** page.

≡ Unified Customer Profile for Travellers and Guests on AWS   ▾ ▮▮▮poc_load_test

DOMAIN
▮▮▮*poc_load_test*   🗑 Delete this Domain

NUMBER OF OBJECTS	NUMBER OF PROFILES	INGESTION ERRORS	VALIDATION ERRORS
3446	590	12	0

DATA REPOSITORIES
Source data locations

👤 Traveller Profile Records   ✈ Air Booking   🎿 Clickstream   💬 Customer Service Interactions   👤 Guest Profiles   🏨 Hotel Booking   👤 Passenger Profiles   💲 Stay Revenue

Solution Setting page - S3 buckets

By default, the AWS Glue jobs will not run automatically. You can manually initiate them from the **Settings** page or configure them to run automatically by setting the **skipJobRun** parameter to `false` when deploying the template.

> ℹ **Note**
>
> Updating the Glue job capacity via the console is not currently supported.

Any ingestion errors display on the **Settings** page.

INGESTIONS ERRORS
Errors reported during the data ingestion process

◀◀ Prev   0-10   Next ▶▶   🔄

Timestamp	Error Type	Business Object	Message	Action
2023-08-14 17:02:32	accp_ingestion_error	hotel_booking	Due to multiple objects trying to map to the same keys we were unable to assign this object to a profile within the allotted time	👁 🗑
2023-08-14 17:02:22	accp_ingestion_error	hotel_booking	Due to multiple objects trying to map to the same keys we were unable to assign this object to a profile within the allotted time	👁 🗑
2023-08-14 17:02:20	accp_ingestion_error	hotel_loyalty	Due to multiple objects trying to map to the same keys we were unable to assign this object to a profile within the allotted time	👁 🗑
2023-08-14 17:01:53	accp_ingestion_error	phone_history	Due to multiple objects trying to map to the same keys we were unable to assign this object to a profile within the allotted time	👁 🗑
2023-08-14 17:00:19	ACCP Change Processing Error	Unknown	AccessDenied: Access Denied status code: 403, request id: H0XPC641EQE7YMGR, host id: 2E5fYa5dRMYE3JzbvtlyH11MSUJ9f6lv7s8i7Z5IfU8TNb/PanBRWcatWxhR29hBWAtAT1IcU+6JpT/kUDkdSA==	👁 🗑
2023-08-14 17:00:14	ACCP Change Processing Error	Unknown	AccessDenied: Access Denied status code: 403, request id: 8571SZY1BWNDWMHA, host id: WiGmw4AIcycfOBW+mH2Vd3ZTgtX+6ngAc6Yeh1b9Lx74iP841k+DSdqn7/oqqly68L2tICqlp/dLWZIDhpI61A==	👁 🗑
2023-08-14 17:00:08	ACCP Change Processing Error	Unknown	AccessDenied: Access Denied status code: 403, request id: 7CP5J2W77E7RB4AD, host id: CCH4tudYeImIotUTYTEfrC4tRRCm06UqwIJHxmMNFUb9PwqQB3BgqOIaw8OEcJttt21XyIe8xK5W6ZyLTIAkVQ==	👁 🗑
2023-08-14 16:58:39	error_parsing_error		Unknown SQS record type with Body '{}' and attributes map[]. Error: Unknown record type (no 'UcpErrorType' or 'Domain' attribute	👁 🗑
2023-08-14 16:28:49	accp_ingestion_error	hotel_booking	Due to multiple objects trying to map to the same keys we were unable to assign this object to a profile within the allotted time	👁 🗑
2023-08-14 16:27:51	accp_ingestion_error	hotel_booking	Due to multiple objects trying to map to the same keys we were unable to assign this object to a profile within the allotted time	👁 🗑

Solution Settings page - ingestion errors

Search, retrieve, and merge profiles using the front end

Once data has been successfully ingested into the solution, the number of profiles and profile objects will appear in the settings screen as illustrated below.

Profiles can then be searched by the following criteria:

- Last name

- Email

- Phone number

- Traveler ID (the ID you pass to the solution)

- Customer profile ID (the ID of the Traveler360 profile created)

> **ⓘ Note**
>
> The search criteria must be an exact match.

When the search returns results, select the **Details** button to access the profile details page view to view the full profile.

The profile details page allow you to view all the data available for this profile and compare and merge duplicate profiles. Duplicate profiles found using the AI-matching mechanism are shown on the right panel along with any data ingestion errors that could have impacted this profile data.

Selecting on the duplicate profiles opens a modal window that compares the profile data and allow customers to merge the profiles together. The merge process is described in details in the Amazon Connect Customer profiles documentation.

Customize the profile details page with dynamic URL

On the profile details page, any data element can appear as a selectable link. This allows you to link the systems of records for a data object (for example, link the Central reservation system reservation details page to a booking ID or the loyalty system to a loyalty ID. This allows users like customer service agents to quickly navigate multiple systems from a single traveler360 page.

To configure selectable elements, go to the setting page and create a URL using the following pattern:

```
https://<system-url>/{{records_field_name}}/.../{{records_field_name }}}
```

where `records_field_name` is the name of the field in the same record.

For example, the template:

```
https://central-res.com/{{ hotelCode}}/{{ bookingId}}
```

will render https://central-res.com/ABCDEF/12345679 and allow to user to navigate to that URL.

Search, retrieve, and merge profiles using the API

All operations available from the user interface are available directly using the API Gateway API created for this solution. This Postman project describes in details the structure of the input, output, and all REST API endpoints provided with this solution.

This allows you to integrate the unified profiles data to any downstream system such as your customer service agent workspace.

Monitoring the solution

The solution provides two main monitoring views.

The frontend settings view

This view is located in the frontend deployed by the solution. This view lists all ingestion errors and the status of Glue jobs and Appflow runs. It also provides high-level metrics on the profile count and convenient access to the source data buckets. Below are a couple of examples:

Top of the frontend of Unified Customer Profile for Travelers and Guests on AWS view

JOBS
Glue Jobs for data transformation and Movements

▶ Run All

| Job Name	Last Run	Status
⬈ clickstreamJobprod	*Never Ran*	not_running
⬈ pax-profileJobprod	*Never Ran*	not_running
⬈ customer_service_interactionJobprod	*Never Ran*	not_running
⬈ hotel-stayJobprod	*Never Ran*	not_running
⬈ hotel-bookingJobprod	*Never Ran*	not_running
⬈ guest-profileJobprod	*Never Ran*	not_running
⬈ air_bookingJobprod	*Never Ran*	not_running

FLOWS
Appflow integration flows to Amazon Connect Customer profile

Business Object	Status	Execution Rule	Last Scheduled Run	Last On Demand Run	Action
air_booking	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:04:46	Successful	Not Run	Run
air_loyalty	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:13:51	Successful	Not Run	Run
clickstream	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:19:54	Successful	Not Run	Run
customer_service_interaction	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:38:04	Successful	Not Run	Run
email_history	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:07:48	Successful	Not Run	Run
guest_profile	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:22:56	Successful	Not Run	Run
hotel_booking	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:28:58	Successful	Not Run	Run
hotel_loyalty	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:25:57	Successful	Not Run	Run
hotel_stay_revenue_items	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:35:03	Successful	Not Run	Run
loyalty_transaction	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:16:52	Successful	Not Run	Run
pax_profile	Active	Scheduled on rate(1hours) starting Mon Aug 14 21:32:00	Successful	Not Run	Run

Bottom of the frontend of Unified Customer Profile for Travelers and Guests on AWS view

The CloudWatch dashboard

The solution creates a custom CloudWatch dashboard with all the key components of the solution to monitor (Kinesis stream traffic, API calls and latency, error queues, ...). The dashboard is accessible from the CloudWatch console, search for: `unified-traveler-profile-prod-`*`<region>`*.
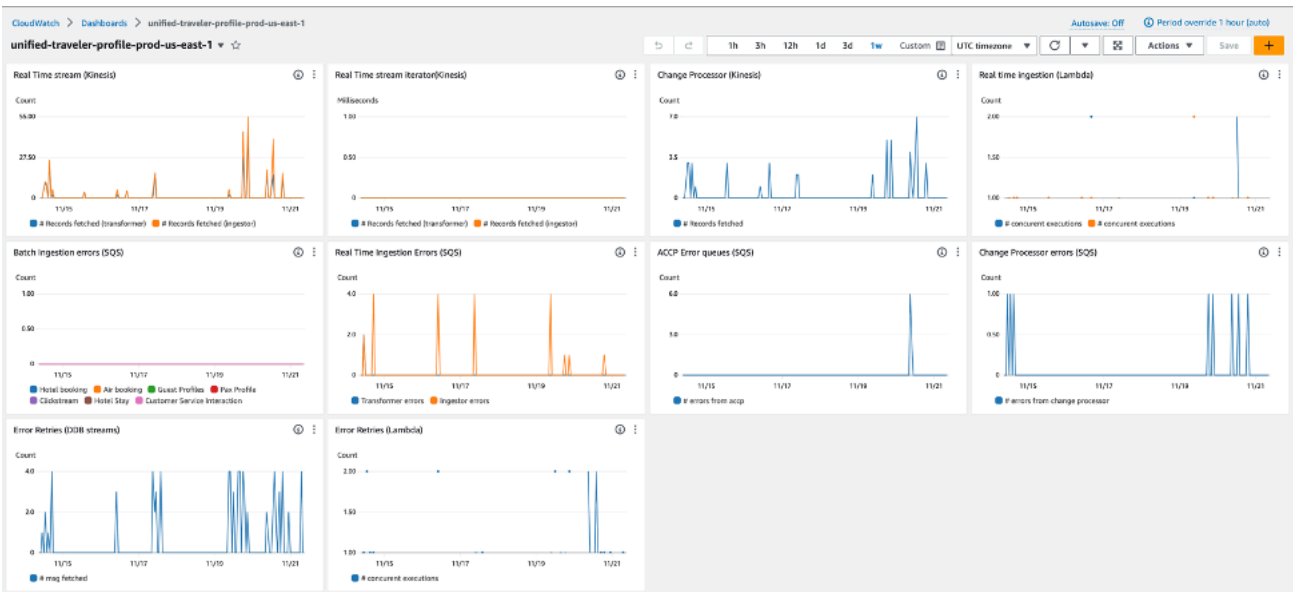
Example CloudWatch dashboard created by the solution

Subscribing to profile change events

The solution creates a dedicated EventBridge eventbus where all the changes made to a customer profile are propagated in real time. Find it in the EventBridge console by searching for the `ucp-traveller-changes-` prefix.

Since version 1.1.0, the solution allows users to receive "MERGED" events. These events occur when two profiles are being merged. Contact your account team to get access to this preview feature.

> **ⓘ Note**
>
> The solution sends the full traveler profile for every batch of changes occurring withing the same few seconds. This allows bus subscribers to use the full profile data to engage the traveler in real time using a downstream system.

Querying the data using Amazon Athena

Since version 1.1.0, the solution creates an AWS Glue table preconfigured with the traveler schema. This allows you to easily perform advanced queries on profile data using Amazon Athena.

The `ucp_traveler_prod` Glue table is partitioned by domain. New partitions are added to the table when domain are created.

The AWS Glue table allows you to [create dashboards](#) with Amazon QuickSight.
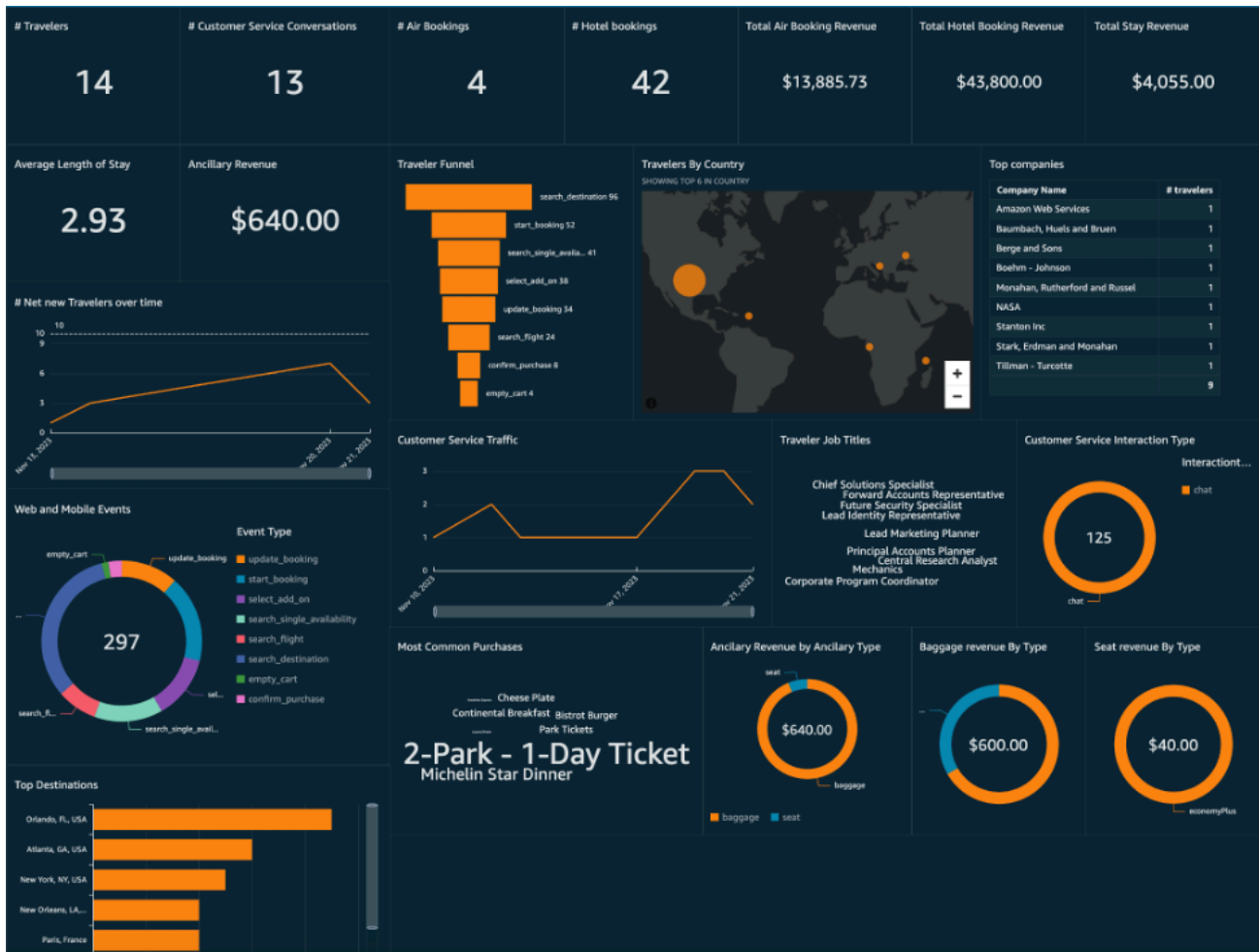
QuickSight dashboard

Visualize the data using QuickSight

Follow this [tutorial](#) to visualize the data ingested in Amazon S3 cluster using QuickSight.

Integrating third-party identity resolution

The solution propagates all matches found by deterministic and AI-based identity-resolution processes to a DynamoDB table. The table shows the schema.

Primary key		Attributes	
domain_sourceProfileId (pk)	match_targetProfileId (sk)	Score	targetProfileID

Primary key		Attributes	
domain_<source_profile_id	match_<target_profile_id>	0.9961272307258183	<targetProfileId>

Any additional processes that produces matches with the same schema in DynamoDB will natively be supported by the solution. Customers can use AWS partners like Amperity or their own batch fuzzy-matching logic to augment the solution's capability accordingly.

> ⓘ **Note**
>
> The ID provided here is the Amazon Connect customer profile ID (available in the **connectId** field of the traveler object).

Integrating with the Travel and Hospitality Application Connectors Catalog on AWS

To integrate the solution with the Travel and Hospitality Connector Catalog on AWS, follow the steps below:

1. To verify if the Travel and Hospitality Application Connectors Catalog on AWS solution is deployed on your account (prerequisite), go to the front-end Settings screen and locate the Connector section. If the button says **Configure**, then the solution is correctly deployed and you proceed to Step 2. If the button says **Deploy**, follow the instructions in the Travel and Hospitality Application Connectors Catalog on AWS solution Implementation Guide to deploy the solution first.

CONECTORS
The Travel and Hospitality connector catalog on AWS solution is currently deployed on your account.

🔗Configure

2. Locate the Amazon S3 bucket used to output the connectors data. For details, see the Travel and Hospitality Application Connectors Catalog on AWS.

3. Input the Amazon S3 bucket name as a parameter of the CloudFormation template for this solution. For details, see [Deploy the solution](#) section.

4. When the deployment or update completes, go back to the Settings screen of the Unified Profiles for Travelers and Guests on AWS solution front end and select on the same button in Step 1 that should now display as **Link**.

5. Choose your domain name and select **Save**.

After these steps, all traffic flowing through the Travel and Hospitality Application Connectors Catalog on AWS solution will be ingested by this solution.

Developer guide

This section provides the source code for the solution.

API reference

This [Postman project](#) describes in detail about the structure of the input, output, and all REST API endpoints provided with this solution.

This allows you to integrate the UCP data to any downstream system such as your customer service agent workspace.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution, pointers to related resources, 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 Unified Profiles for Travelers and Guests on AWS deployment

- **Timestamp** - Data-collection timestamp

- **Example: Instance Data** - Count of the state and type of instances that are managed by the EC2 Scheduler in each AWS Region

Usage Data

General solution usage information for data ingestion and web app usage.

Example Data:

```
Records Ingested Real Time {duration: 100ms, record_count: 10}
Web App {usecase: SearchProfile, duration: 100ms, status: "success"}
Web App {usecase: CreateDomain, duration 100ms, status: "failed", error_step:
 "validate", error: "domain already exists"}
```

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

1. Download the `ucp.template` the section called "AWS CloudFormation template" to your local hard drive.

2. Open the AWS CloudFormation template with a text editor.

3. Modify the AWS CloudFormation template mapping section from:

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

to:

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

4. Sign in to the 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**, select **Choose file** and select the edited template from your local drive.

8. Choose **Next** and follow the steps in Launch the stack in the Deploy the solution section of this guide.

Related AWS Solution

The solution integrates with the Travel and Hospitality Application Connectors Catalog on AWS solution to provide off-the-shelf data feeds from eleven hotel property management systems (PMS) and clickstream events from Tealium Customer Data Platform (CDP).

Contributors

- Owen Brady

- Akash Garg

- Geoffroy Rollat

- Cassidy Neal

Revisions

Date	Change
August 2023	Initial release
October 2023	Release v1.0.1: Updated package versions to resolve security vulnerabilities.
November 2023	Documentation update: Added Confirm cost tags associated with the solution to the Monitoring the solution with AWS Service Catalog AppRegistry section.
December 2023	Release v1.1.0: Performance improvements and schema updates. Additional CloudFormation template parameters for better control of Kinesis Data Streams. Removed a section on setting up the Customer Profiles export stream, which is now automatically done by the solution.
February 2024	Release v1.1.1: Update of the Python and NodeJS Lambda runtimes.

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.

Unified Profiles for Travelers and Guests on AWS is licensed under the terms of the Apache License Version 2.0.