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: October 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 CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template

Features and benefits

The solution provides the following features:

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 solution frontend. 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 a dedicated interaction store built for travel and hospitality brands on top of Amazon Aurora. The data is then replicated in low latency caches (Amazon Connect Customer Profiles and DynamoDB). 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.

Versions 2.0.0 and above provide a new data privacy feature allowing travel brands to locate and erase traveler profiles across all solution data stores (Amazon S3, DynamoDB, Amazon Connect Customer Profile) to facilitate compliance with privacy laws like General Data Protection Regulation (GDPR).

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 Running SQL queries 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.

Data lineage

With version 2.0.0 and above, the solution’s interaction store stores the lineage of every profile through all identity resolution merge activities. This allows customers to identify and correct inaccurate identity resolution results from either AI false-positive results or user error.

Every traveler interaction is associated a confidence factor based on how it came to the current profile (AI-identity resolution, Rule based, or Manual merges).

Customers can undo inaccurate merges and restore profiles while preserving the full lineage.

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.

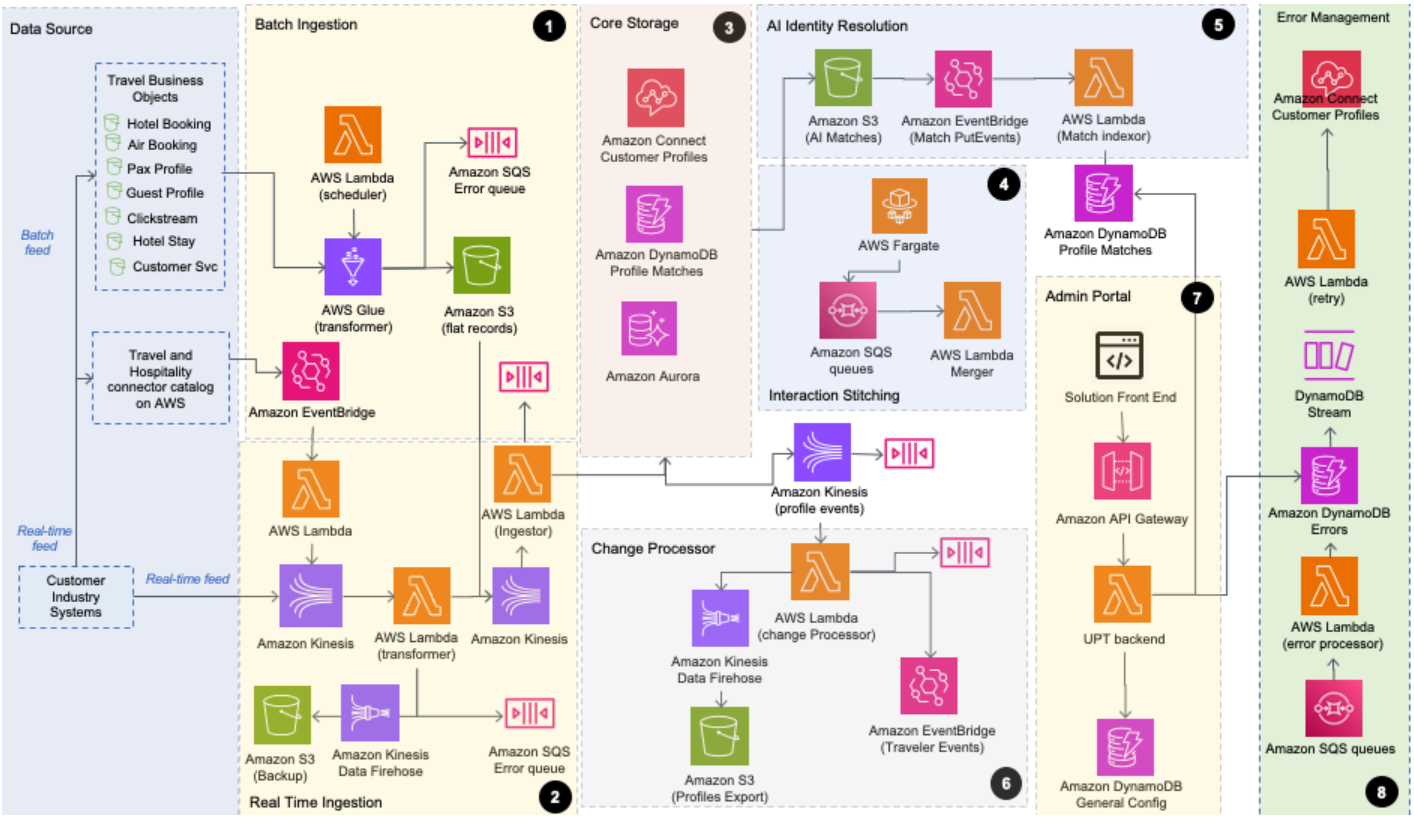
> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary.

Architecture overview

This section provides a reference implementation architecture diagram with description and AWS Well-Architected design considerations.

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 solution architecture is composed of eight core modules:

1. **A batch ingestion module** allows customers to send data to Amazon Simple Storage Service (Amazon S3), run Extract, Transform and Load (ETL) jobs with AWS Glue, and ingest data in highly parallelized Lambda functions into the solution's core storage module (interaction store).
2. **A real-time ingestion module** allows customers to send real-time data to Amazon Kinesis Data Streams. Data is then transformed using the same transformation function used by the Glue jobs in the batch ingestion module and ingested into the solution's interaction store.
3. **The core storage module (interaction store)** is an Amazon Aurora Serverless cluster that ingests transformed records (interactions) in a custom-built transactional data store while maintaining consistency and data lineage. This module also optionally caches every customer profile into an Amazon DynamoDB table for low latency access.
4. **The interaction stitching module** indexes every interaction into a dedicated Amazon Aurora table and runs lightweight match queries against the index table to identify interactions that match a set of rules (configured in the solution frontend). An AWS Fargate cluster allows you to start a large number of parallel task that fetch interactions from Aurora and replays them against the index whenever rules change. This module sends interactions identified as a match of the same profile to a merge queue for consolidation.
5. **The connect and AI-identity resolution module** receives relevant consolidated profiles from core storage and caches them in Amazon Connect Customer Profiles low latency store. Identity resolution is run by Amazon Connect Customer Profiles which identifies matches and sends them to an S3 bucket to be processed back by the solution (and merged above a threshold). Once in Amazon Connect Customer Profiles, customers can leverage the Amazon Connect integration feature.
6. **The change processor module** receives all changes made to a profile and sends them downstream to an S3 bucket, and Amazon EventBridge allows customers to easily integrate downstream applications.
7. **The administration portal module** provides a user interface and a REST API allowing customers to configure the solution and search and retrieve profiles. This web portal can be integrated into Amazon Connect third-party application framework to be used by customer service agents.
8. **An error management module** is composed of 13 Amazon Simple Queue Service (Amazon SQS) queues strategically placed throughout the solution upstream and downstream data path to gather all errors (transformation, ingestion, downstream event creations, and identity resolution). These errors are processed and indexed into a DynamoDB table and available in a sorted real-time log through the admin portal which makes the solution easy to monitor. The solution also includes a custom Amazon CloudWatch dashboard and a set of custom metrics using CloudWatch embedded metric format.

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 provides an advanced permission feature allowing the Administrator to enforce least privilege for users using functional rules. The solution provides a privacy search and purge feature allowing customers to locate PIIs in every storage component within the solution and delete them.

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.

From v2.0.0 and above, the solution's interaction store can use the Aurora backup and point-in-time recovery features to improve resilience from previous versions. After restoring the Aurora cluster to a point-in-time version, customers can repopulate the low latency caches by starting a cache-rebuild job from the solution's user interface.

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.

Starting in version 2.0.0, the solution's interaction store provides consistent sub-second ingestion performance at 5600 interactions per second. Amazon Connect Customer Profiles profile retrieves function that have been benchmarked at 50ms, and DynamoDB provides single-digit millisecond first-byte latency for the most performance-critical use cases (like web personalization).

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.

Traveler Interaction Store

The solution deploys an Aurora Serverless v2 cluster used as primary high-performance storage for all traveler interactions. The database stores interaction history and data lineage, and ensures consistency and accurate profile and interaction count. Profiles are stitched together in real time using interaction-level rule-based matching configurable from the solution user interface.

Two optional downstream caches (Amazon Connect Customer Profiles and DynamoDB) store profiles for low latency access. Users choosing Amazon Connect Customer Profiles also benefit from additional features such as AI-identity resolution using the Amazon Connect integration.

The data is ingested and stored in the interaction store 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 (Aurora table, Amazon Connect Customer Profiles Domains, DynamoDB Table, and 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, deleted, and merged, a real-time stream is created out of the solution interaction store 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 solution user interface.

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.

Privacy search and delete

A dedicated screen in the solution user interface allows customers to search for a list of traveler IDs and locate records in the solution storage components (S3, DynamoDB, Aurora). This allows for centralized access and deletion to all PIIs associated to a profile which facilitates the enforcement of privacy laws.

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. The front-end also allows customers to access the privacy search feature to facilitate GDPR compliance or other regulation.

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 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.

AWS services in this solution

AWS service	Description
Amazon Aurora	**Core.** Store traveler interaction and index them for rule-based identity resolution.
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, and integrates with Amazon Connect.
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 processin g from data ingested prior to the solution deployment.
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

AWS service	Description
	dedicated event bus is created by the solution and it receives events in the event of traveler data changes.
AWS Fargate	**Core.**Runs containerized tasks allowing you to reindex large amounts of data in parallel for identity resolution when rules change and repopulates caches during interaction store restores.
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.
Amazon Bedrock	**Optional.** Amazon Bedrock is used to generate the optional summary of traveler profiles using the Anthropic Claude Foundational model.

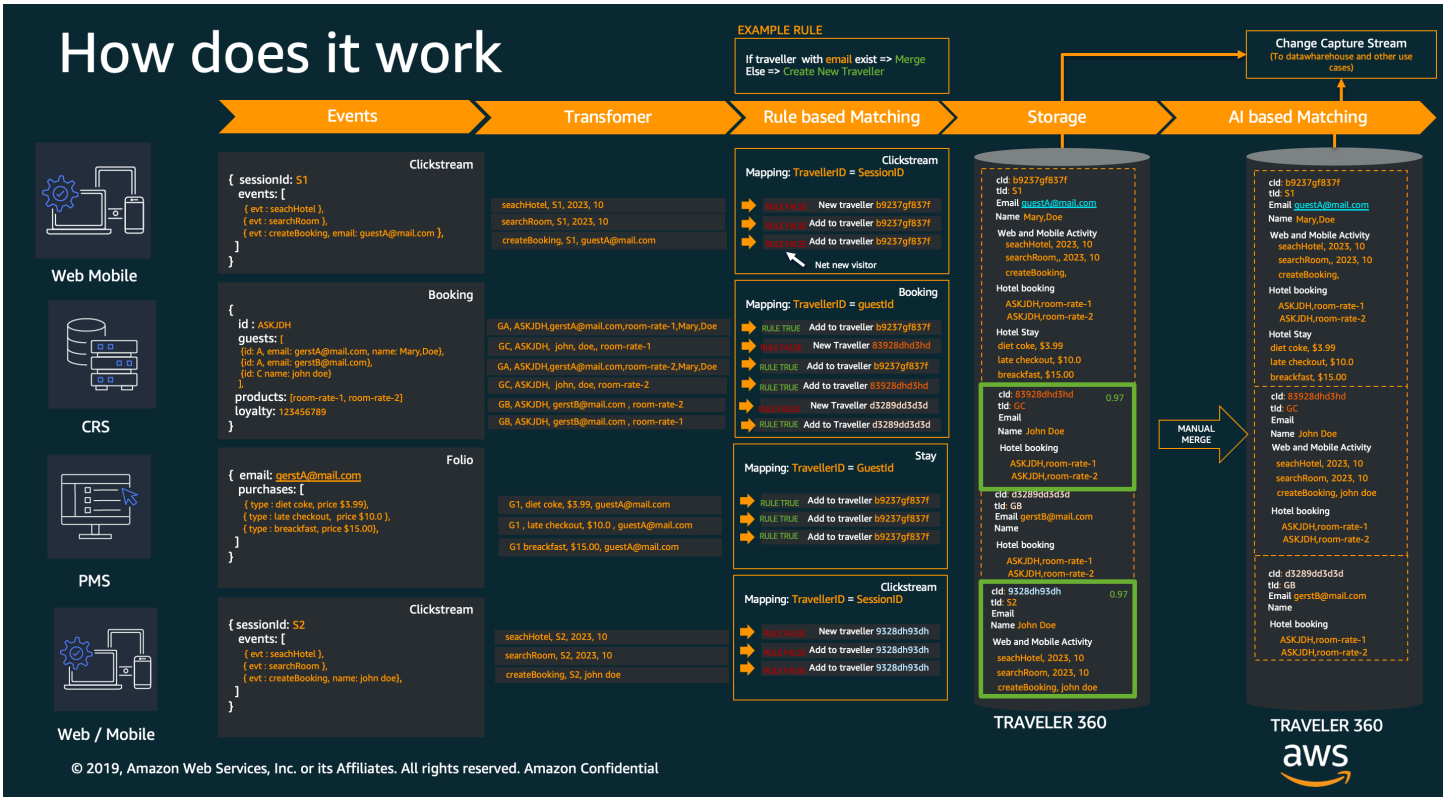
How this solution works

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.

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 for users enabling the Amazon Connect Customer Profile storage and AI-Identity Resolution, and $0.0015 per traveler profile per year without.

> **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 with AI-Identity Resolution [USD]	Cost per year without AI-Identity Resolution [USD]
100,000 profiles	$3,800.00	$150.00
1,000,000 profiles	$38,000.00	$1,500.00
10,000,000 profiles	$380,000.00	$15,000.00

Number of profiles	Cost per year with AI-Identity Resolution [USD]	Cost per year without AI-Identity Resolution [USD]
Greater than 10,000,000 profiles	Contact your account team to discuss tiered pricing options.	$150,000.00

Note

95% of the cost associated with this solution relates to data storage components (Amazon Connect Customer Profiles, Aurora, and DynamoDB). 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, Privacy Deletion, and Generative AI Prompt Update. 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:

Profile search and display permissions

Profile search and display permissions

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

Use case permissions

This solution comes with features that can be broadly divided into two categories:

Domain agnostic features - These features are domain independent. These may include settings which are shared across domains or being able to configure domains.

These features include:

- CreateDomain and DeleteDomain - Ability to create and delete a domain.
- ConfigHyperlink - Create hyperlink mappings for attributes.
- RunGlueJobs - Running Glue jobs for different business objects.
- ClearAllErrors - Clearing errors observed during ingestion.

Domain specific features - These features are tied to a domain. These may include ruleset settings, performing profile operations, and configuring gen-AI summary generation for a domain.

These features include:

- SearchProfile and DeleteProfile - Ability to search and delete profiles.
- MergeProfile and UnmergeProfile - Ability to merge and unmerge profiles.
- SaveRuleSet and ActivateRuleSet –SaveRuleSet saves a draft rule set that has been modified. ActivateRuleSet makes that rule set active for rule-based matching.
- ConfigureGenAi - Configuring traveler summary generation settings.

Permissions

Each feature has a binary representation associated to it, known as the Permission. Every binary string will have only one bit set as 1 except for the admin permission.

For example, let's take a look at the following permission set:

Here each permission has a single bit set as 1. A user can access a feature if their permission string also has the same bit set as 1. The DomainAdmin permission has all bits set as 1, meaning this permission can access all features in a domain.

```
{
           "SearchProfilePermission" :
           "0000000000000001",
           "DeleteProfilePermission" :
           "0000000000000010",
           "MergeProfilePermission" :
           "0000000000000100",
           "UnmergeProfilePermission" :
           "0000000000001000",
           "CreateDomainPermission" :
           "0000000000010000",
           "DeleteDomainPermission":
           "0000000000100000",
           "ConfigureGenAiPermission" :
           "0000000001000000",
           "ConfigHyperlinkPermission" :
           "0000000010000000",
           "SaveRuleSetPermission" :
           "0000000100000000",
           "EditRuleSetPermission" :
           "0000001000000000",
           "ActivateRuleSetPermission" :
```

```
            "0000010000000000",
            "RunGlueJobsPermission" :
            "0000100000000000",
            "ClearAllErrorsPermission" :
            "0001000000000000",
            "RebuildCachePermission" :
            "0010000000000000",
            "DomainAdmin": "1111111111111111"
}
```

Cognito groups

For each Cognito group starting with prefix app, you will see a hex value in the group name. Converting the hex value into a binary string will result in a set of 1's and 0's. The positions of 1's indicate which features are accessible by the group.

Although you can create your own groups, the solution provides some predefined group.

Predefined Cognito groups

Decoding the group name: `app-{group_type}-{permission_group_name}/{hex_code}`

The group names start with the prefix app

- group_type - can be global (for domain agnostic) or {domain_name} (for domain specific)
- permission_group_name - indicates the type of permissions controled by the group
- hex_code - defines the features accessible by the group

After deploying the solution, you will see six predefined Cognito groups:

Domain agnostic permission groups -These are created once, after deploying the solution. There are two groups in this category:

Global-DomainManagerRole

1. Add a user to this group if you would like them to create and delete domains
2. Hex value 30 translates to 0000 0000 0011 0000
3. The same value can be computed by performing a bitwise OR of CreateDomainPermission and OR of DeleteDomainPermission

Global-SettingsRole

1. Add a user to this group if you would like them to run Glue jobs, clear errors, or add hyperlink mappings
2. Hex value 1880 translates to 0001 1000 1000 0000
3. The same value can be computed by performing a bitwise OR of RunGlueJobsPermission, ClearAllErrorsPermission, and ConfigHyperlinkPermission.

DomainSettingsRole

1. Add a user to this group if you would like them to manage domain settings like configuring Gen-AI prompt or rebuilding cache
2. Hex value 2040 translates to 0010 0000 0100 0000
3. The same value can be computed by performing a bitwise OR of ConfigureGenAiPermission or OR of RebuildCachePermission.

ProfileManagerRole

1. Add a user to this group if you would like them to manage profile data for a domain
2. Hex value F translates to 0000 0000 0000 1111
3. The same value can be computed by performing a bitwise OR of SearchProfilePermission, DeleteProfilePermission, MergeProfilePermission, and UnmergeProfilePermission

RuleSetManagerRole

1. Add a user to this group if you would like them to manage rule sets for a domain
2. Hex value 700 translates to 0000 0111 0000 0000
3. The same value can be computed by performing a bitwise OR of SaveRuleSetPermission, EditRuleSetPermission, and ActivateRuleSetPermission.

DomainAdminRole

1. Add a user to this group if you would like them to access all the features
2. Hex value FFFF translates to 1111 1111 1111 1111
3. The same value can be computed by performing a bitwise OR of all the permissions

User defined groups

To create your own groups for providing feature level access:

1. Sign in to the AWS Management Console and go to Amazon Cognito.
2. In the navigation pane, choose **User pools** and select your user pool.
3. Select the Groups tab and choose **Create group.**
4. While building the group name, follow these steps:
   a. The group name should be of format:`app-{group_type}-{permission_group_name}/{hex_code}`
   b. If you want the group to be used across all domain, group_type should be global
   c. If you want the group to be associated to a specific domain, group_type should match the name of the domain
   d. `permission_group_name` can be something meaningful to your use case
   e. Include the / after the permission_group_name
   f. Hex code value – omit the base prefix (0x) while using big-endian (MSB first), and do not use padding. The value should not exceed the width allocated (16bits or 4 hex characters).
   g. Using the list of permissions as a reference (see Permissions), build a binary string by performing a bitwise OR for the features needing access
   h. For example, to create the group with access to features - SearchProfilePermission and CreateDomainPermission
      i. SearchProfilePermission: 0000000000000001
      ii. CreateDomainPermission: 0000000000010000
      iii. Bitwise OR: 0000000000010001
      iv. This translates to hex value: 17
      v. The group name could be: app-global-customGroup/17

> **Note**
>
> If your group name fails to meet the above guidelines, the solution will not decode correctly which will result in loss of access to features.

5. Once you have the group name, select **Create group**.

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.

The following service quotas are relevant for this solution:

- Parallel Lambda execution
- Number of Fargate tasks

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.

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)	

Note

For users that would like to opt into the Bedrock, please review the Supported AWS Regions in the Amazon Bedrock guide.

Deploy the solution

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

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 through 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 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

[Optional] Enable Anthropic Claude 3.0 Sonnet model in Bedrock

Starting in v2.0.0 of this solution, customers can enable profile summarization using Bedrock. The Anthropic Claude 3.0 Sonnet model can be enabled in the Bedrock console (see the Amazon Bedrock Anthropic Claude models for more details).

[Optional] Create an Amazon Connect instance

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 (AI Matching) can be monitored using the Amazon Connect Customer Profiles console. This console is only accessible by creating (or using an existing) 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.
industryConnectorBucketName	None	If provided, the solution will set up data replication 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

Parameter	Default	Description
		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.
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`).

Parameter	Default	Description
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.
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`).

Parameter	Default	Description
enableRealtimeBackup	`false`	Enable or disable (default) the real-time backup. If enabled, all messages sent to the solution Kinesis stream will be automatically backed up in S3. Note that Enabling this option requires user to manually locate and delete profiles to comply with privacy laws like GDPR as the privacy search and delete does not include the backup bucket.
customerProfileStorageMode	`true`	Enable profile replication in Amazon Connect Customer profiles. (Note that at least one of the two parameters customerProfileStorageMode and dynamoStorageMode must be set to `true`.)
dynamoStorageMode	`true`	Enable profile replication in DynamoDB. (Note that at least one of the two parameters customerProfileStorageMode and dynamoStorageMode must be set to `true`).

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.

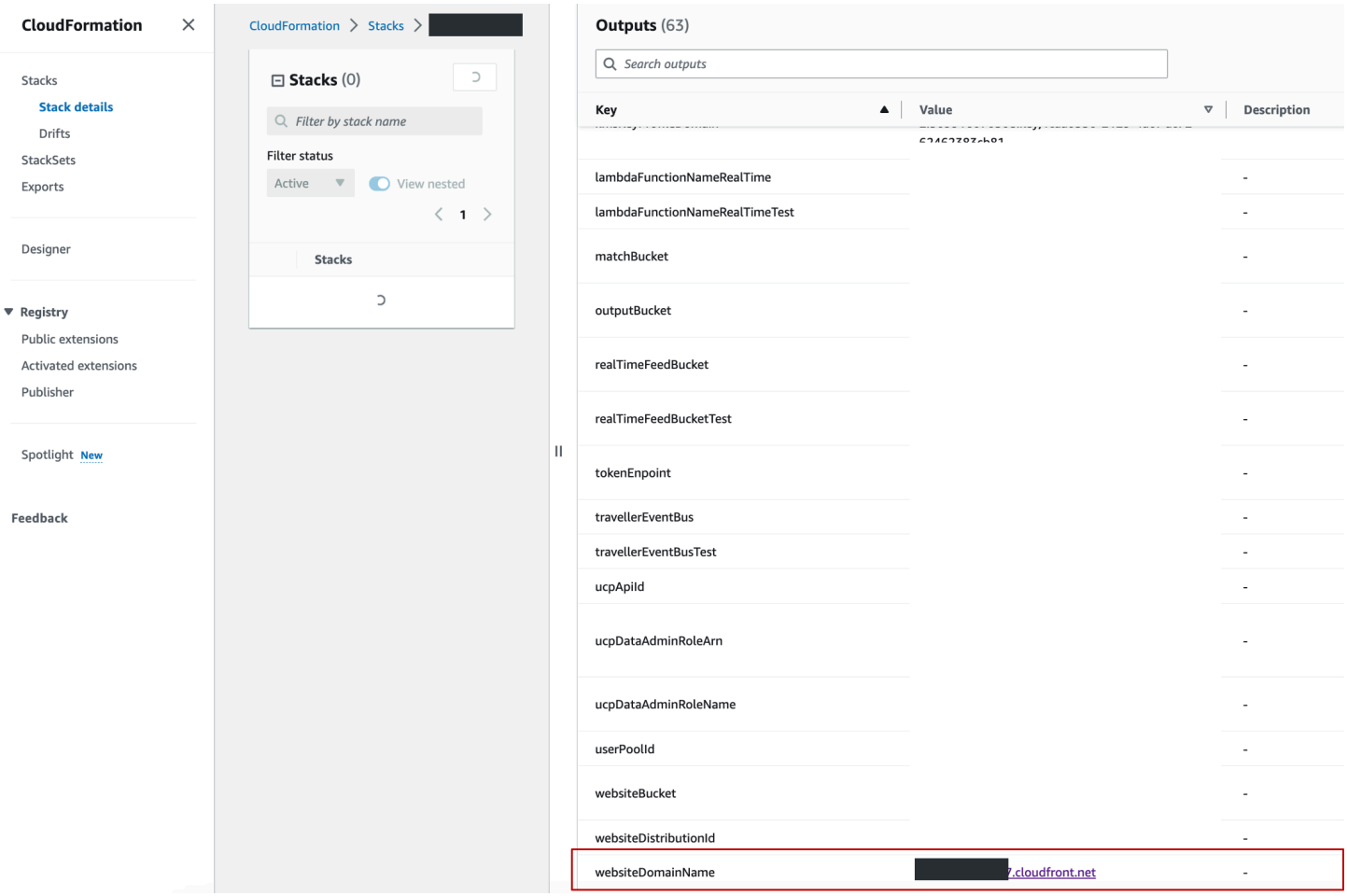
1. Navigate to the Amazon Cognito console with an IAM user that has permissions to create users and groups.
2. Locate the userpool created by the solution (prefixed by `ucp-`).
3. Create an Amazon Cognito user in this userpool for your administrator.

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

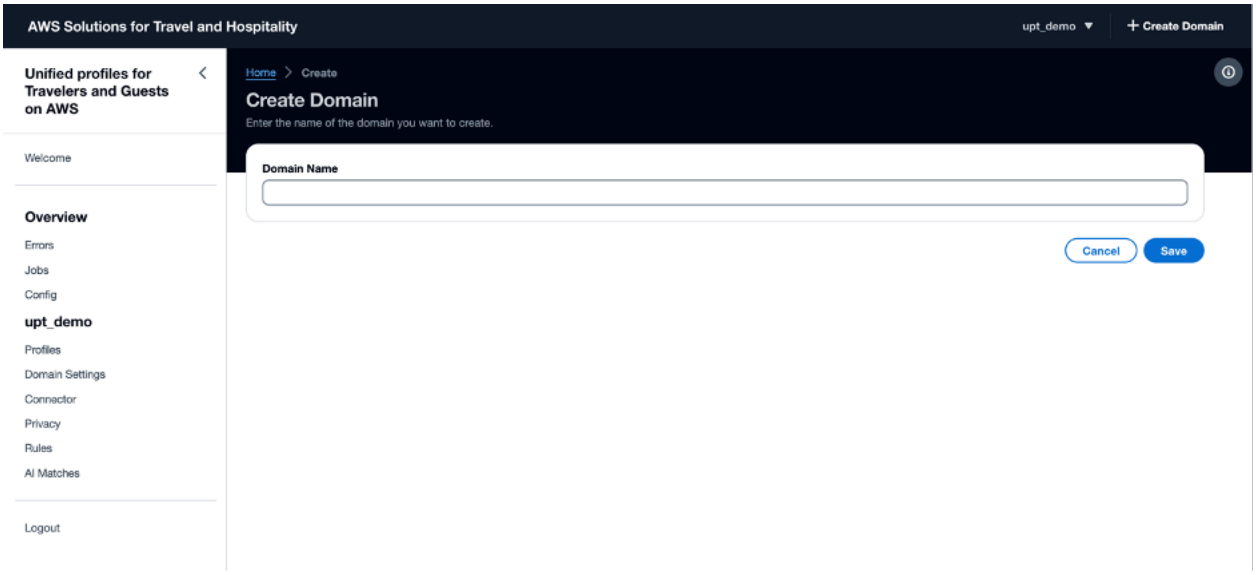
> **Note**
>
> By default, users will be able to login but will not see any traveler data. (Refer to the Amazon Cognito section for steps on how to give user scoped permission to see traveler data.)

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. Select **Create Domain** (located on the upper-right corner of the screen).



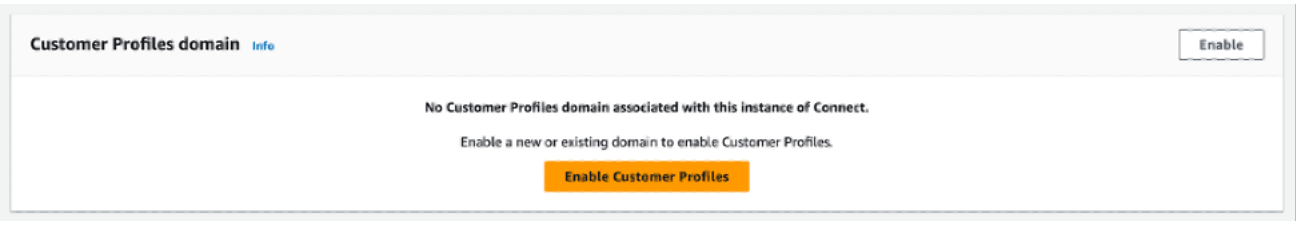
3. Enter a domain name (must follow the snake case convention). Take note of the domain name which will be required for multiple steps.

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.

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

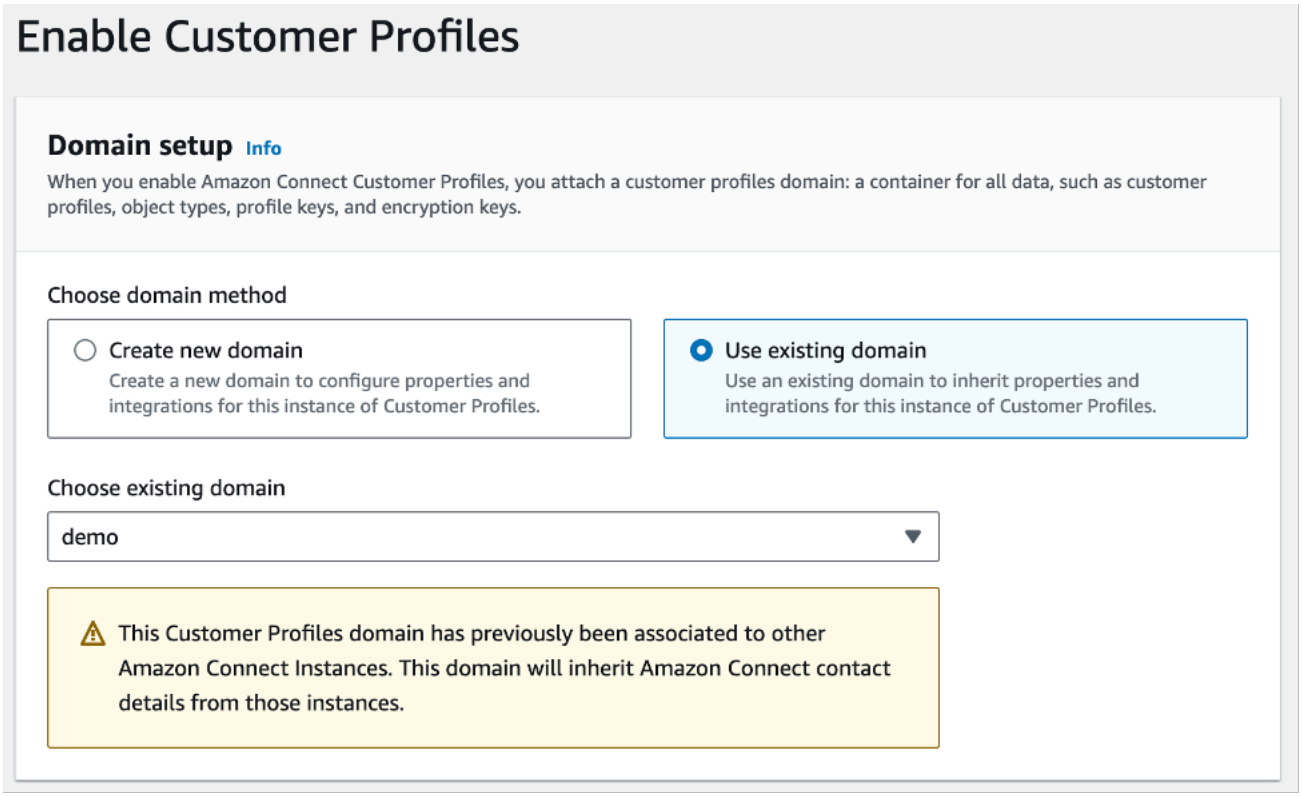
1. Navigate to the Amazon Connect console.
2. Go to Customer Profiles and associate your Amazon Connect Customer Profiles domain with the instance you created.
   a. If you do not have an Amazon Connect instance, create one, then select the instance, and go to the Customer Profiles section.
   b. At 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.

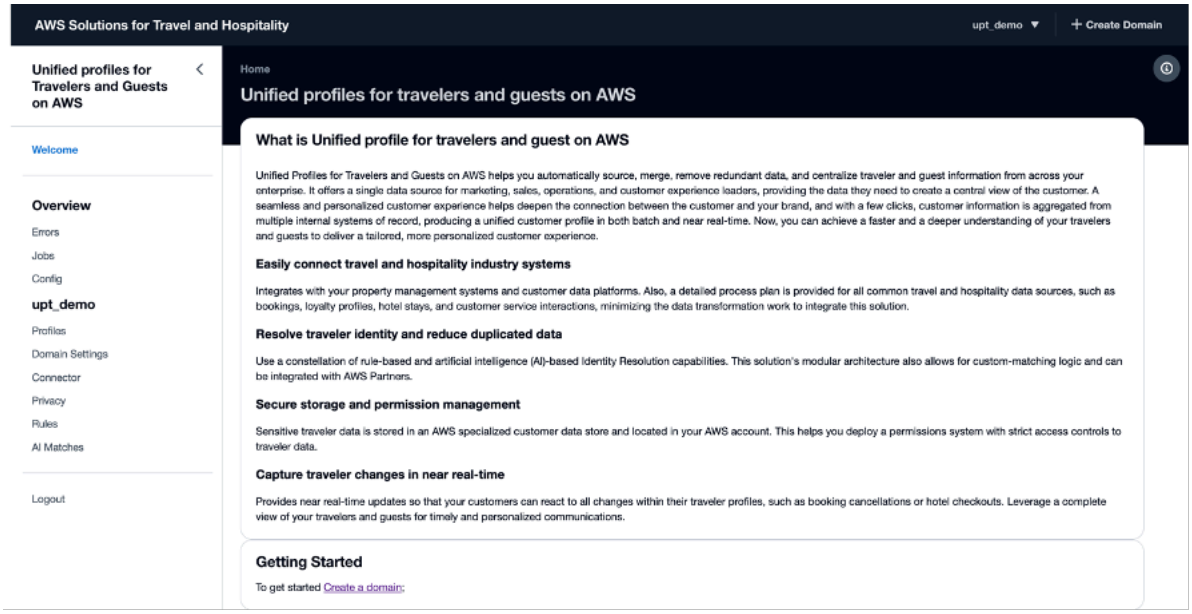
3. Choose Enable, select Use existing domain and select the domain from the list to attach the domain from the section called "Step 3: Create an Amazon Connect Customer Profiles domain".



4. Leave the remaining options as set by the solution and choose **Enable Customer Profiles**. This allows you to configure identity resolution rules. See [Optional] Step 7: Set up AI based Identity Resolution through Amazon Connectfor information about how to set up AI based Identity Resolution.

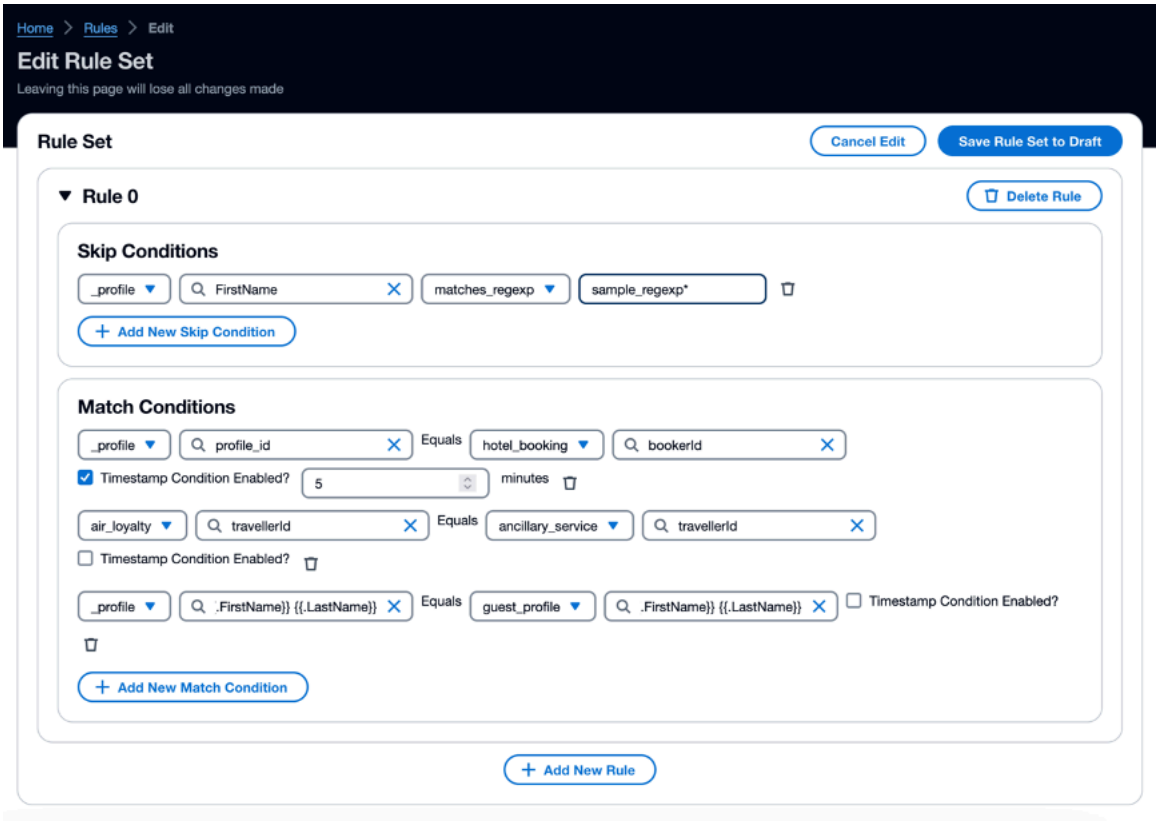
Step 6: Create match and merge rules

1. Go to the solution user interface.



Solution interface

2. Select the **Rule Menu** on the left panel.
3. Choose **Add New Rule**.
4. Create optional Skip condition and Match conditions (more information is provided in the solution contextual help panel).
5. Add other rules and conditions.
6. Select **Save Rules as Draft** (this will save but not activate the rules).
7. Choose the **Draft version** in the solution dropdown list.
8. Select on Activate Draft RuleSet. This last step will initiates an indexing job (you can see the task executing on the Amazon ECS console), and activate real-time rule-base matching.



Edit rule set

[Optional] Step 7: Set up AI based Identity Resolution through Amazon Connect

> **Note**
>
> You must complete [Step 5] Map the Customer Profiles domain to your Amazon Connect instance before performing this step.

1. Navigate to the Amazon Connect console. In the left navigation pane, choose **Customer Profiles**.
2. On the **Customer Profiles** page, go to **Identity Resolution** and choose **Enable Identity Resolution**. Choose **Enable Identity Resolution** to confirm.
3. Navigate to the CloudFormation console, and select **Stacks**.
4. In the **Outputs** tab, identify the S3 match bucket with the **Key** named **matchBucket**, and record its value.

5. In the S3 console, locate the match bucket. In the **Objects** tab, choose **Create folder** and create a folder named **Match**. The AI-based identity resolution CSV output is saved to this folder.
6. Navigate to the Amazon Connect console. On the **Customer Profiles** page, choose **View Identity Resolution**.
7. In **Identity Resolution settings**, select **Machine learning resolution**.
8. On the **Machine learning resolution settings** page, if machine learning resolution is disabled, choose **Enable Machine learning resolution** to enable AI identity resolution. Choose **Enable** to confirm.
9. In **Run Schedule**, choose **Edit** to define the weekly run schedule for machine learning resolution. Select a day and time and choose **Save**.
10. In **Match results location**, choose **Edit.** Select the **Write profile ID matches to Amazon S3** checkbox and specify the S3 bucket location. Then, select the **Match** folder created in step 5 and choose **Choose**.

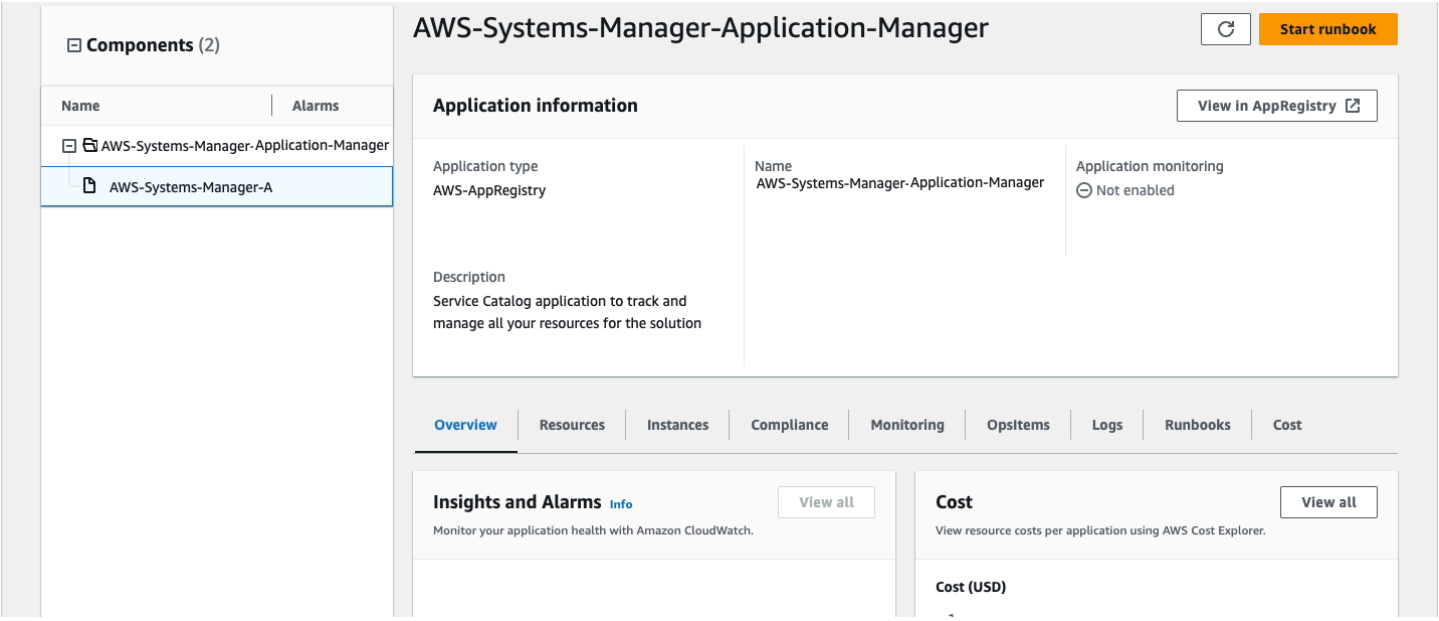
Monitor the solution with Service Catalog AppRegistry

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.



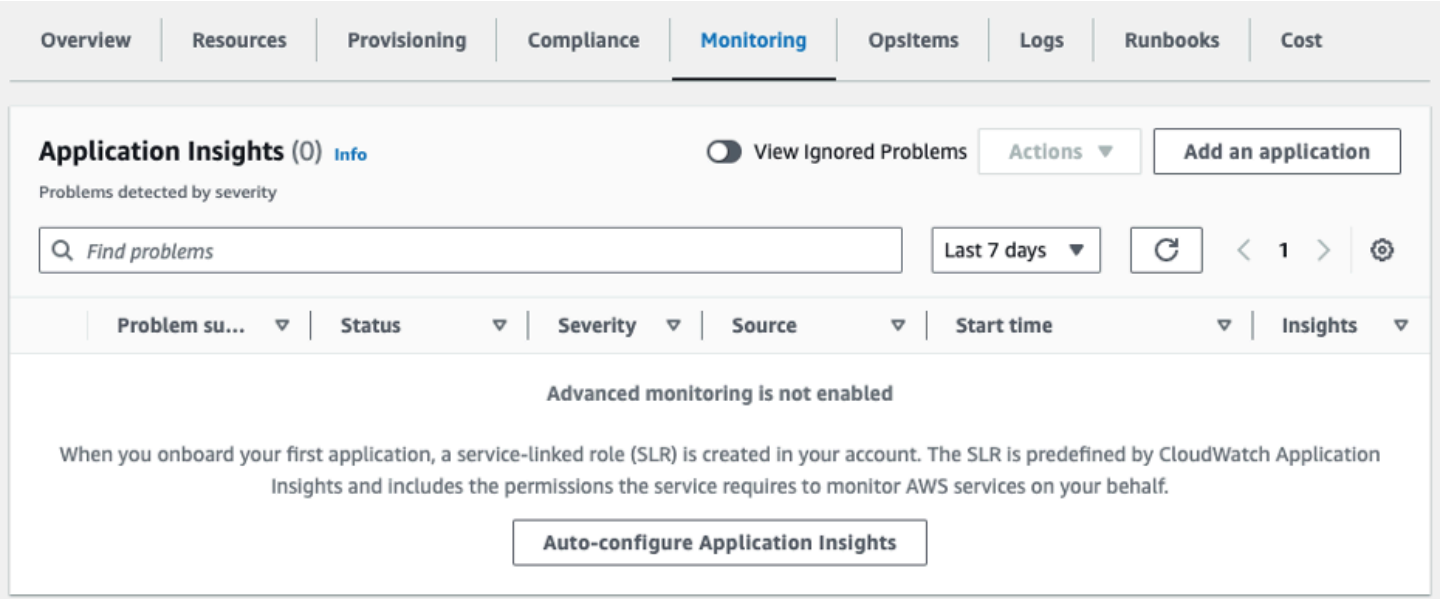
Solution stack in Application Manager

Activate CloudWatch Application Insights

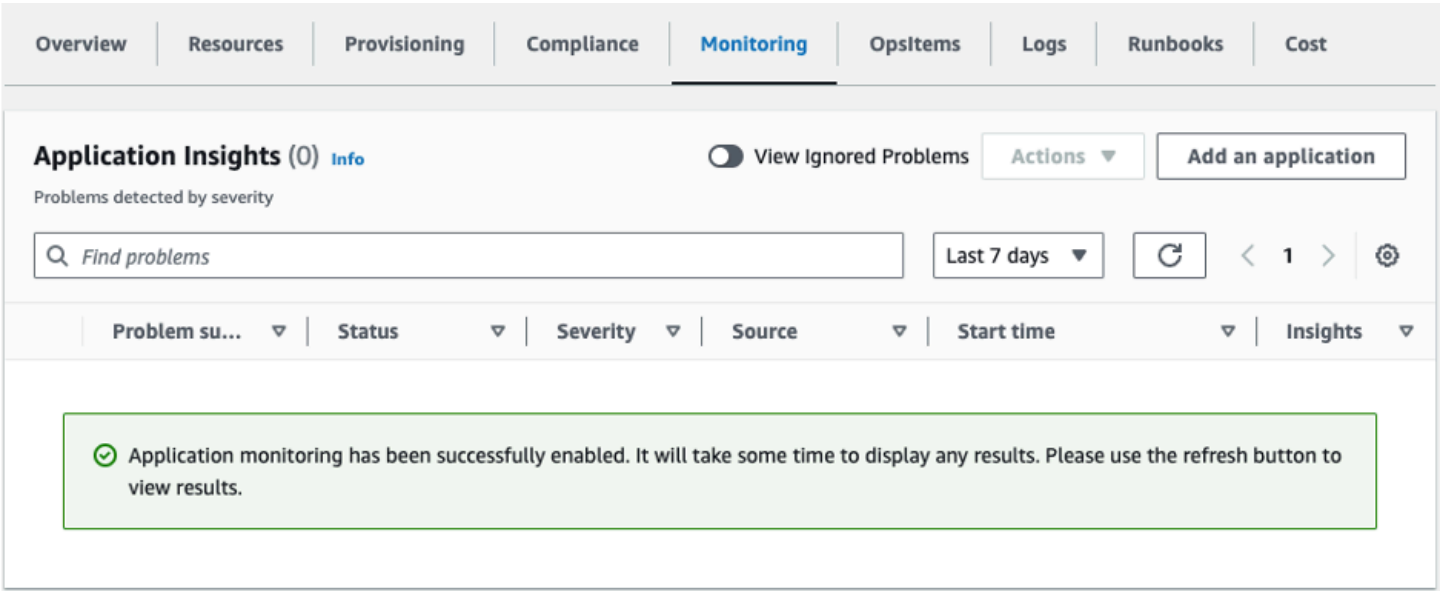
1. Sign in to the Systems Manager console.

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:

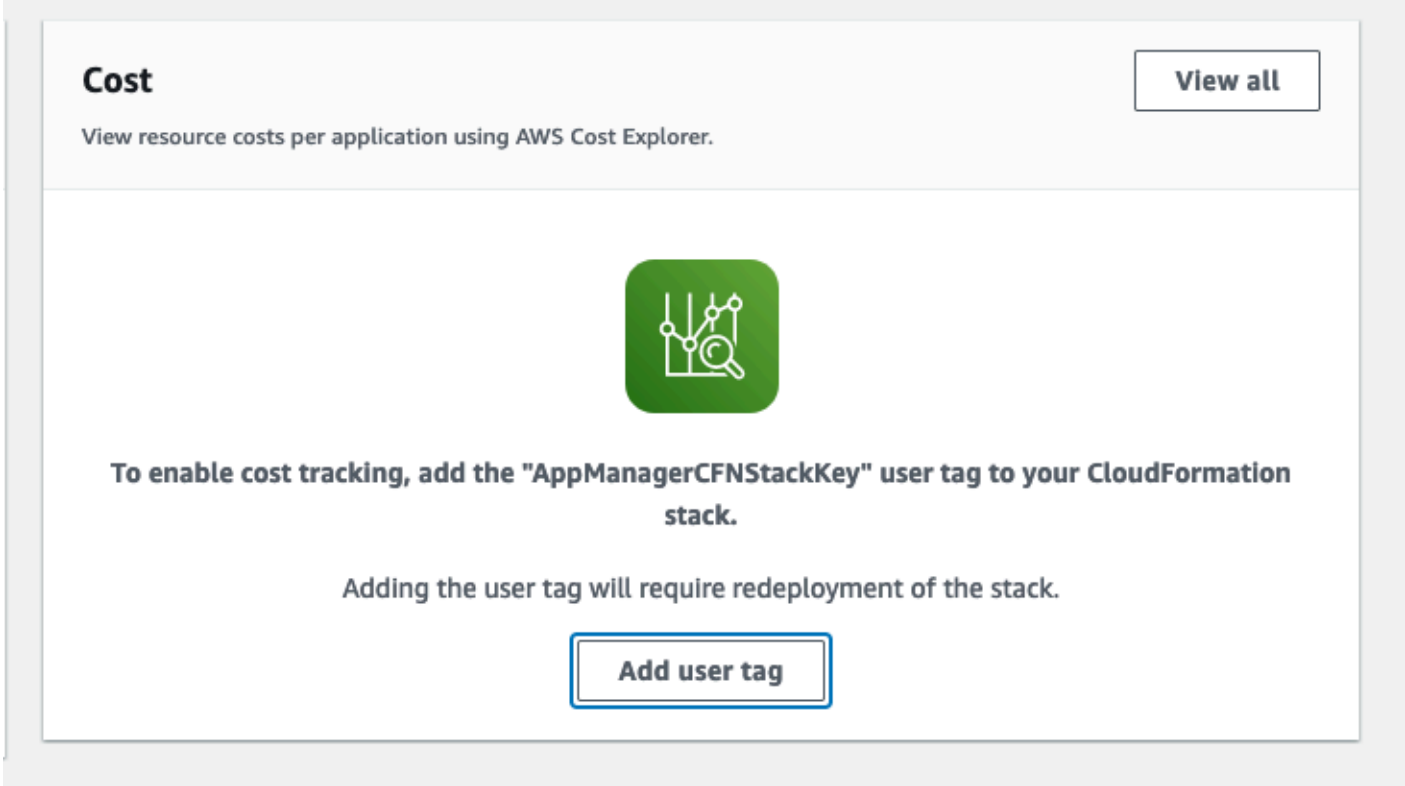


Confirm cost tags associated with the solution

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.

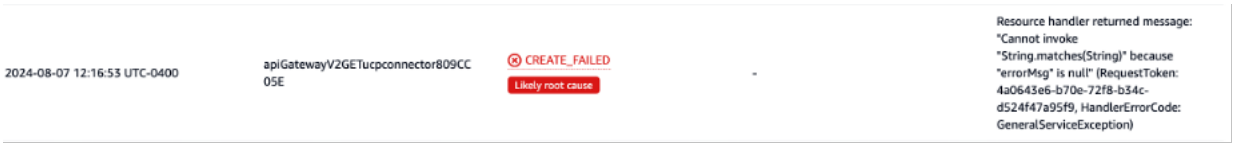
If these instructions don't address your issue, see the Contact AWS Support section for instructions on opening an AWS Support case for this solution.

Problem: Error during deployment

The following error (see example below) can occasionally occur when deploying the solution for the first time. This is a known issue with the API Gateway control plane and can lead to a stack rollback which fails due to S3 bucket not being empty and RDS deletion protection.

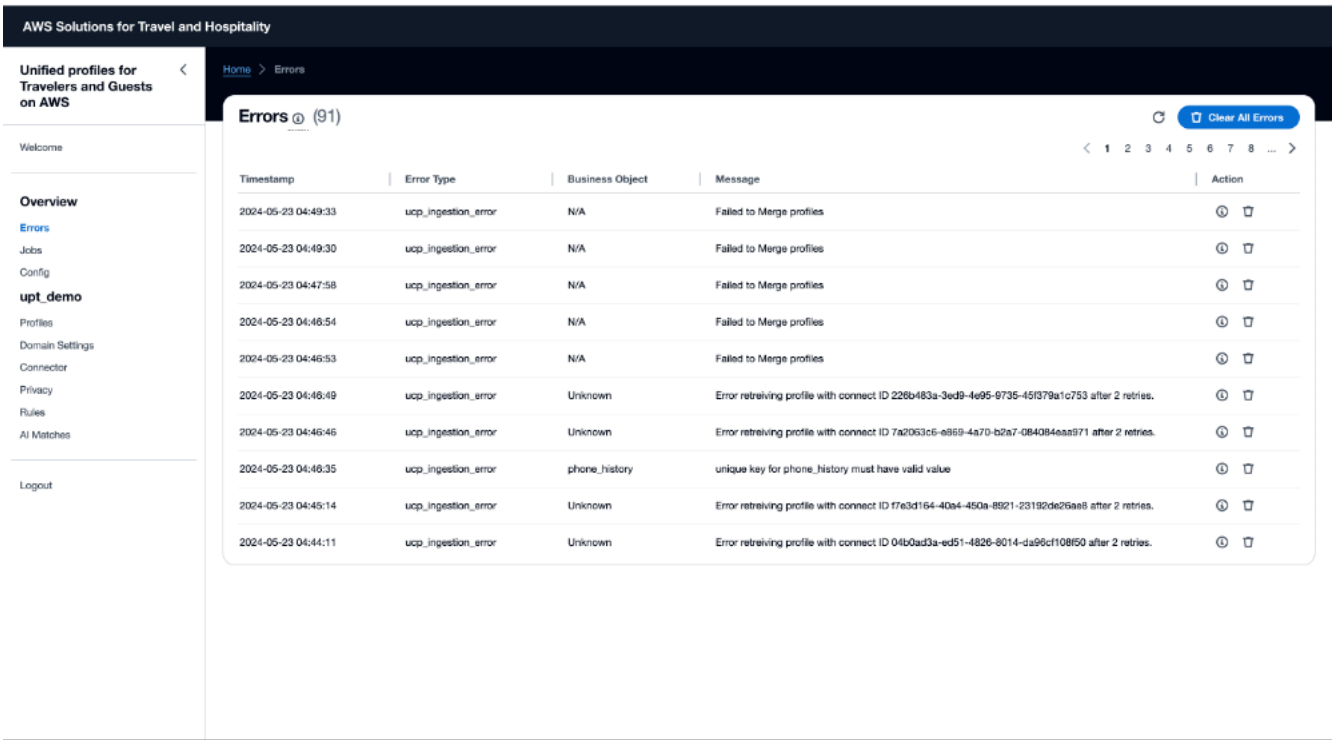
Follow the steps below to fix this issue:

1. Go to the Amazon S3 console, then empty and delete buckets created by the stack.
2. Go to the RDS console and disable deletion protection on the UPT cluster.
3. Return to the CloudFormation console and delete the stack.
4. Launch the stack again. If the error persists, submit a ticket to AWS Support.



Problem: Ingestion errors

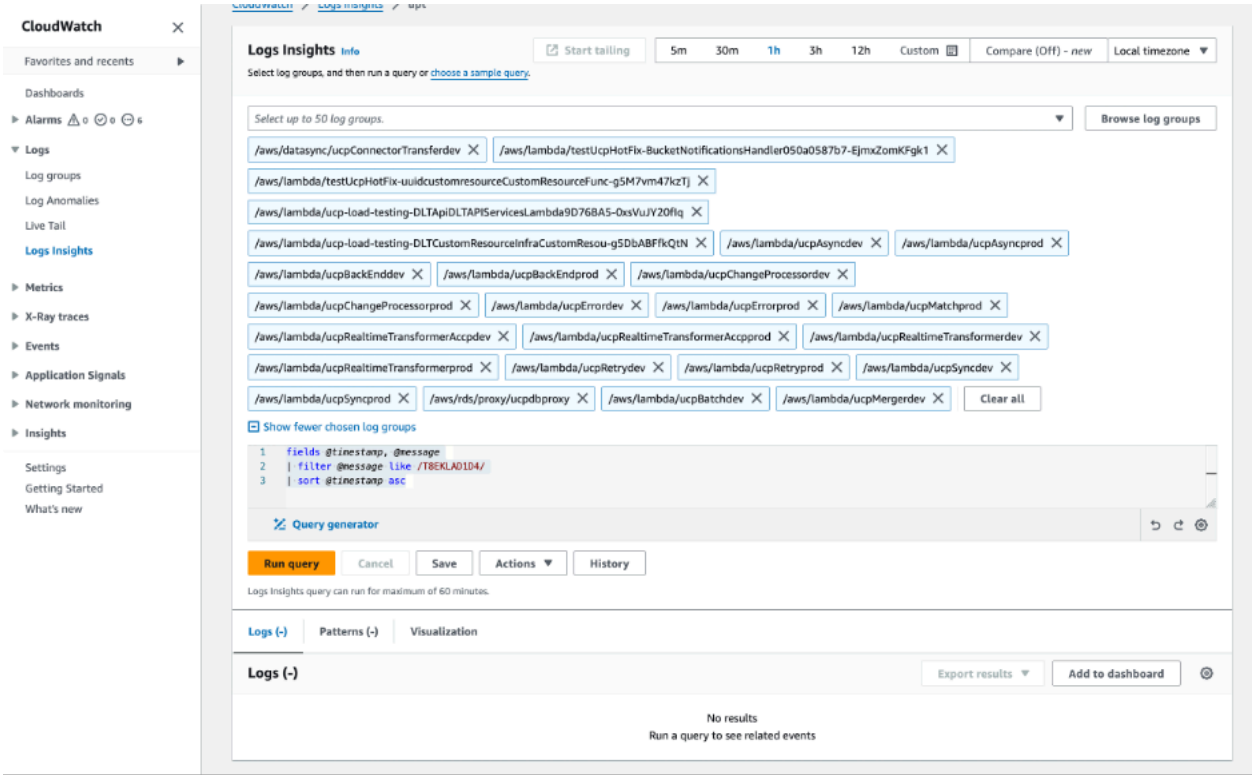
All ingestion errors are ingested into a DynamoDB database and displayed in real time on the **Errors** screen.



Common errors and resolution

Searching for logs in the solution

The solution generates trace IDs (called transaction ID or tx_id) allowing you to easily troubleshoot ingestion and identity resolution errors. We recommend you create and save a CloudWatch Log Insight query in order to search logs efficiently through the solution components. Transaction ID is generally available within the body of the error message presented in the error screen.



Common error types

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”,
  ...
}
```

a database-level error occurred while...

This error occurs when an ingestion into the interaction store fails. To troubleshoot it, access the error body by selecting the information object. Locate the transaction ID, and perform a CloudWatch log insight search as described in the Searching for logs in the solution section.

Unknown SQS record type with Body '{}' and attributes map[]

This error occurs at domain creation when Amazon Connect Customer profiles sends a test message in the error queue to validate access. You can safely disregard it.

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 *sends 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 a flat record. 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 experienc e (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 reservati on.
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,

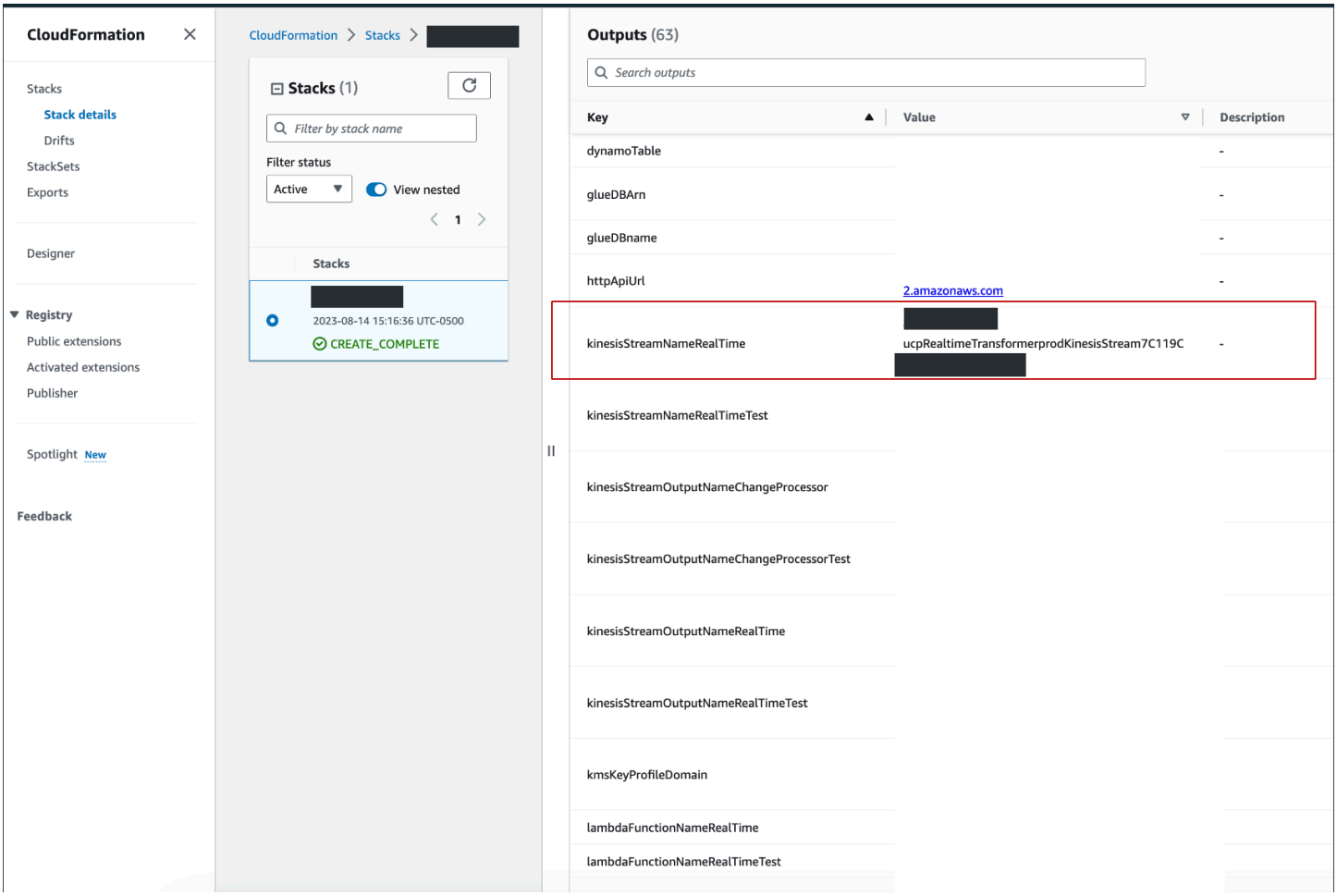
Object name	Fields
	address_mailing_line1, pronoun, address_mailing_line4, address_line2, last_name, language_code, email, object_type, model_version, nationality_name, address_billing_city, address_billing_country, address_business_line1, phone_type, address_business_line4, date_of_birth, address_country,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

Real-time ingestion performance recommendations

By default, the Kinesis streams use the on-demand provisioning mode. This means that Kinesis handles the heavy lifting of scaling the shards of each stream according to the incoming data throughput. If you decide to send a significant amount of traffic to Kinesis in parallel, the Kinesis shards will increase but it may take hours to increase to the necessary amount which will slow down the ingestion process. For a fast, large-scale ingestion, we recommend using the solution's CloudFormation parameters to preprovision the Kinesis shards according to the following table:

Traffic	Transformer Stream shards	Ingestor Stream shards
100 TPS	2	14
1000 TPS	14	70
5000 TPS	70	350

General troubleshooting

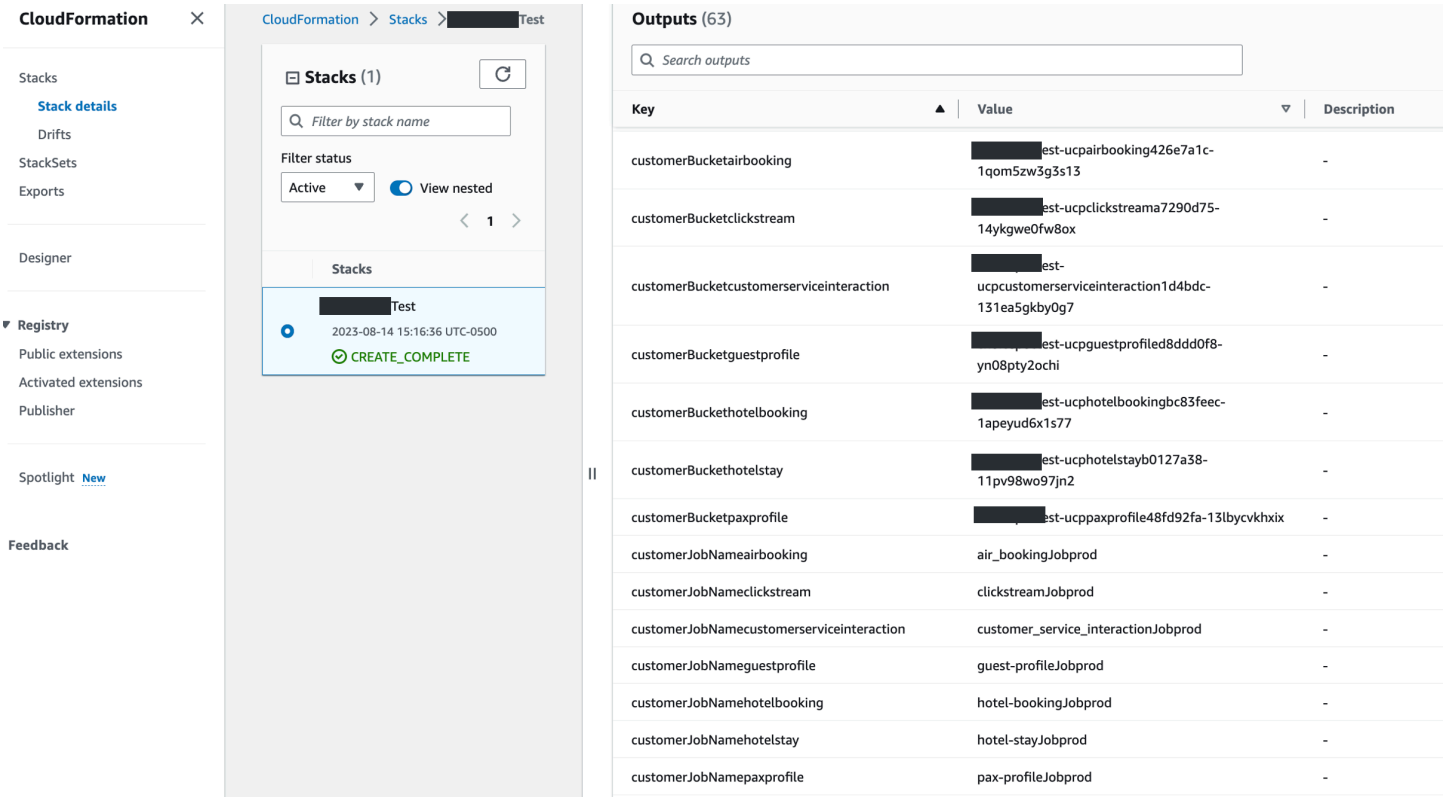
Avoid null types. (For example, emptylist, strings or empty objects are preferred to null types which will rate transformation issues.) Using null types in input business objects will lead to the `NoneType` object is not iterable error viewable on the settings screen.

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:

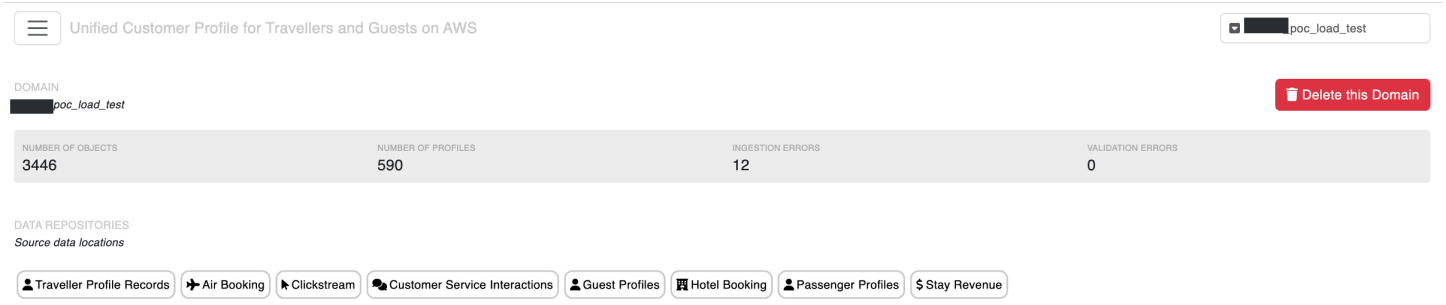
- The root folder must be the ACCP domain name
- The path must contain the lastUpdate timestamp provided in the following format: YYYY/MM/DD
- Objects must be ingested as JSONL (line-break-separated JSON objects where JSON data must not be prettified)

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.



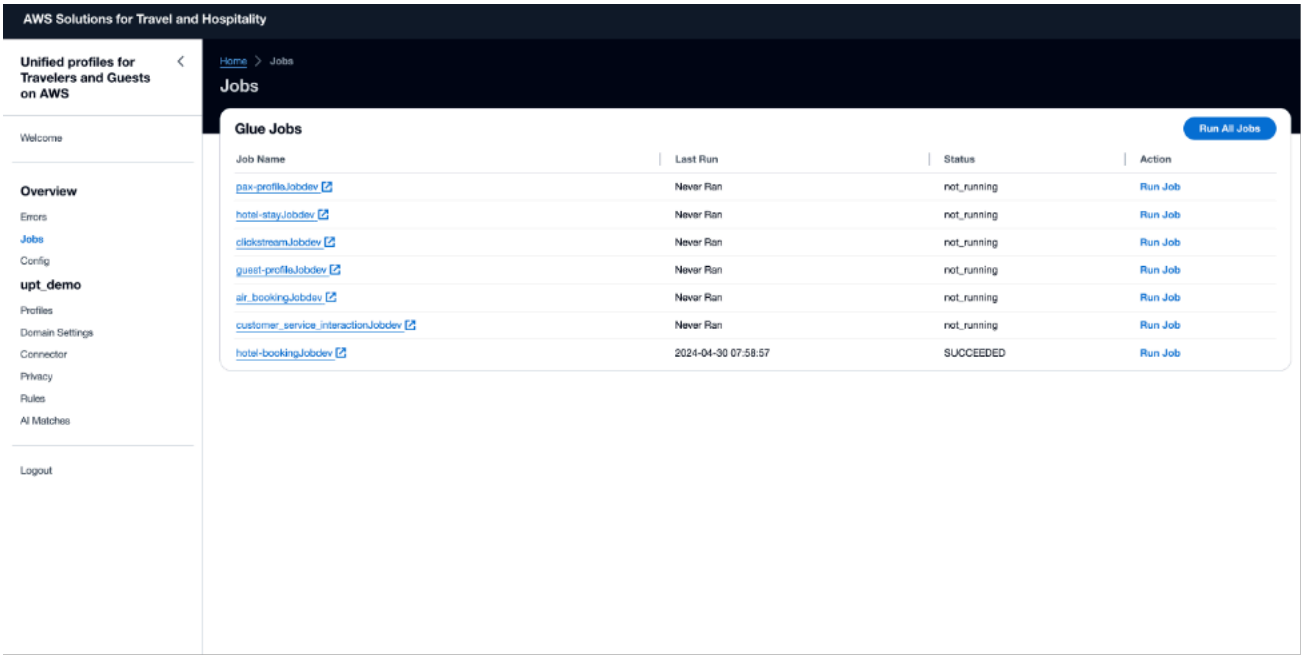
Example of the CloudFormation output

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



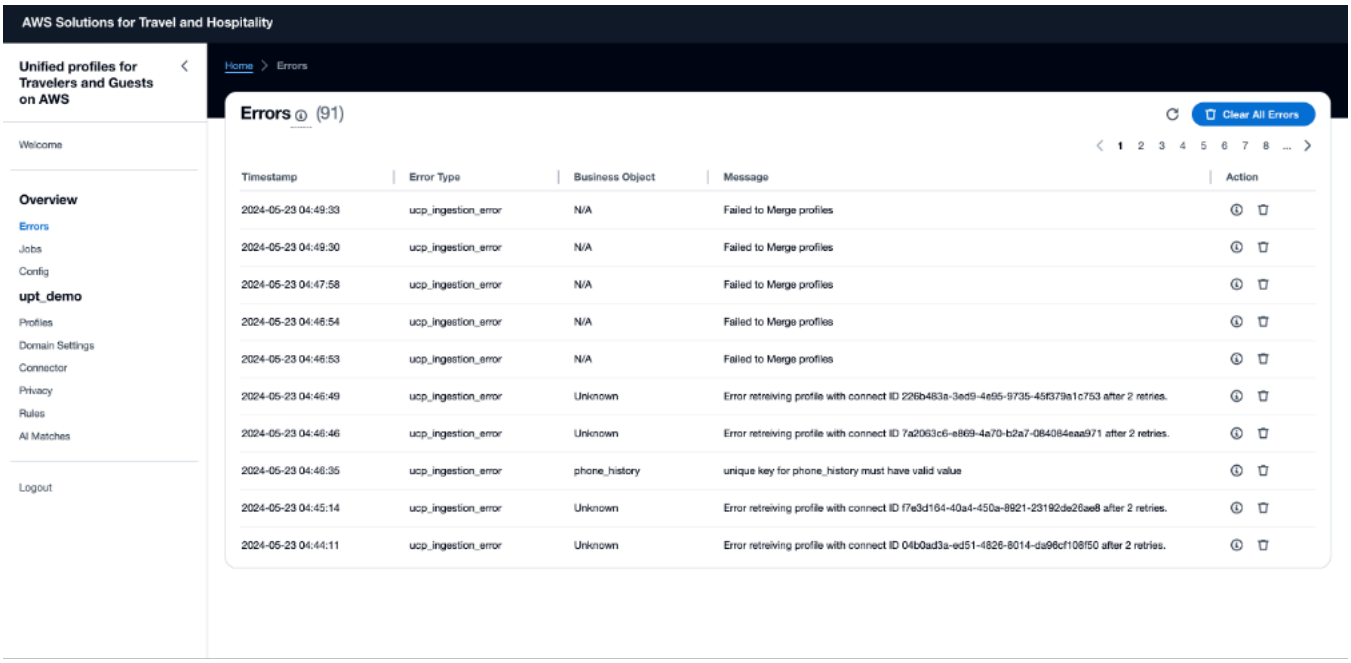
Solution Setting page - S3 buckets

By default, the AWS Glue jobs will not run automatically. You can manually initiate them from the **Settings** page.



Glue jobs

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



Solution Errors page - ingestion errors

Batch ingestion Performance recommendation

Glue Jobs Performance

The Glue jobs are configured with 10DPU default value which allows for ingesting about 1 million large records in less than 20 minutes. For larger amount of data, the DPU amount can be increased from the AWS Glue console. Note that the Glue jobs perform 1-to-many transformation on every business object without the need for shuffling or reducing the data. This makes the transformation process highly parallelizable which means that the job run time will decrease proportionally to the DPU increase.

Interaction Store ingestion performance

In their default configuration, the Glue jobs outputs about 2000-line items into S3 with a total throughput of 4MB/seconds. These files are then processed in real time by Lambda functions. The ingestion process within the Lambda function is parallelized with a 20 parallel thread count maximum. This means a large amount of traffic can be sent to the Aurora cluster. Aurora Serverless, as well as DynamoDB, will scale automatically to meet the demand; however, we recommend to update the DynamoDB table provisioning mode to the following values based on the number of objects ingested:

#Objects to batch ingest	DynamoDB Provisioning (WCU)
1M (ca 20 min)	4000 WCU
10M (ca 2 hours)	40000 WCU
100M (ca 16 hours)	400000 WCU (quota increase necessary)

Follow the tutorial below to update the DynamoDB provisioning mode, see the AWS DynamoDB documentation.

Amazon Connect customer profile ingestion performance

The default TPS limit of Amazon Connect Customer Profiles is 100 TPS (see Amazon Connect qutoas). To enable fast and reliable ingestion performance, Amazon Connect Customer Profiles is decoupled from the integration store using Amazon SQS which is processed by a dedicated

Lambda function. The event processing MAX_PARALLEL_PROCESSING setting can be updated in the solution's CloudFormation parameters to meet the ingestion performance needed. The default values is set to match the 100 TPS limit.

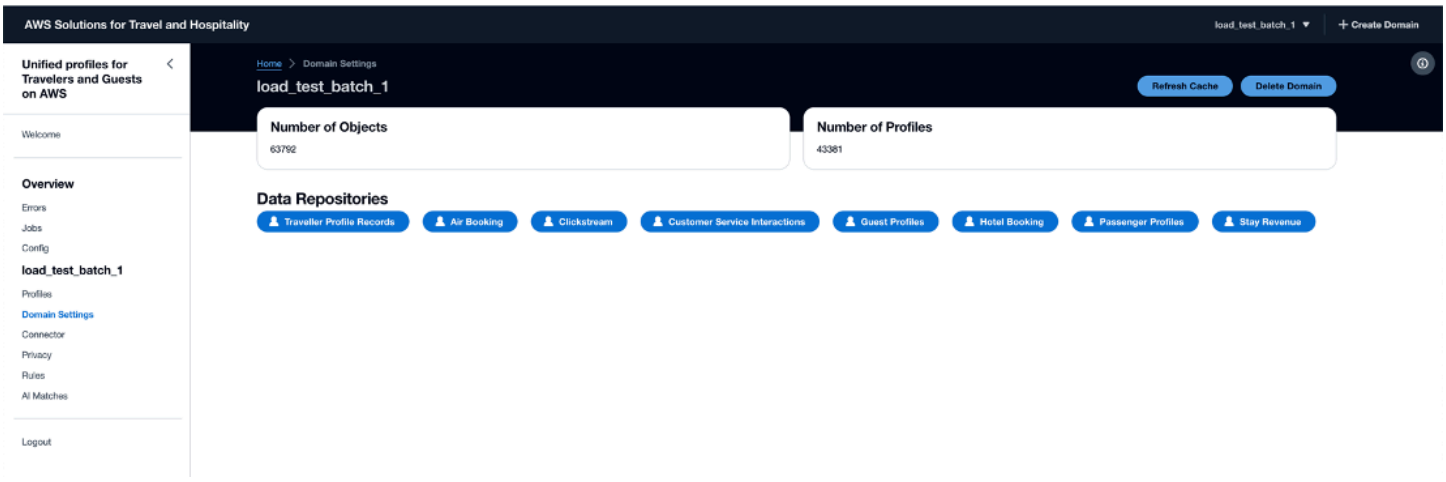
You can request a service limit increase by contacting AWS support. Once granted, update the solution cloudformation parameter according to the following values:

Traffic	Max parallel Lambda processing
100 TPS	2
1000 TPS	20
5000 TPS	200

> **Note**
>
> The solution has not been tested above 5000 TPS in ingestion performance. Contact your account team if you need higher throughput.

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.

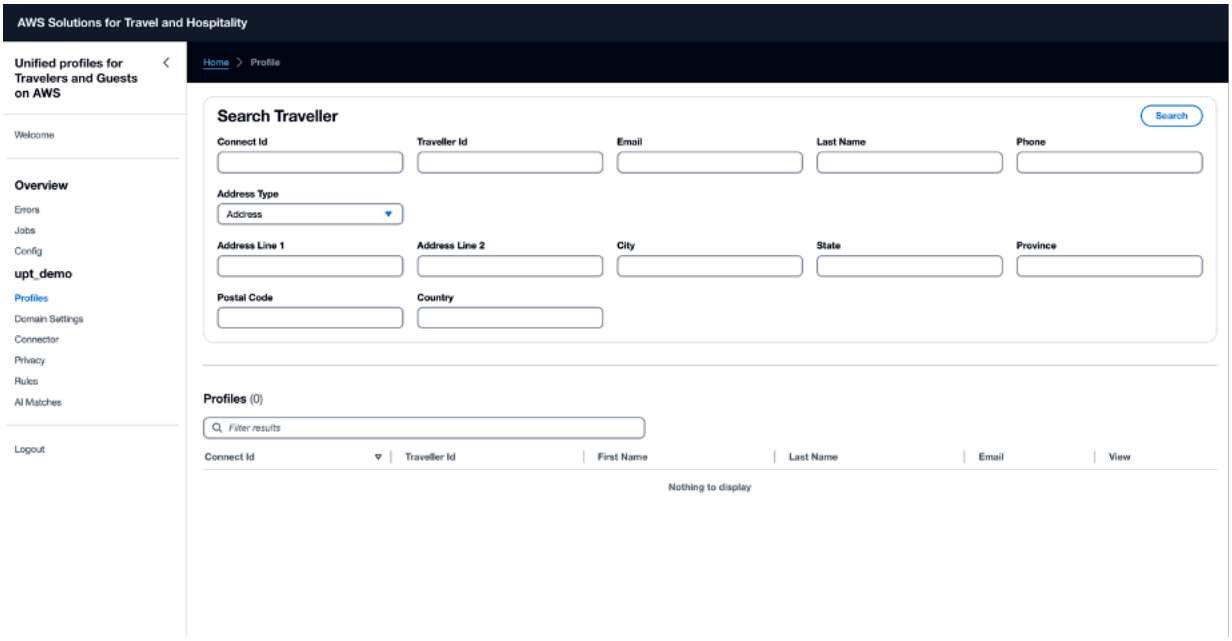


Profiles can then be searched by the following criteria:

- Last name
- Email
- Phone number
- Traveler ID (the ID you pass to the solution)
- Customer profile ID (the ID of the Traveler360 profile created)
- Any Address field

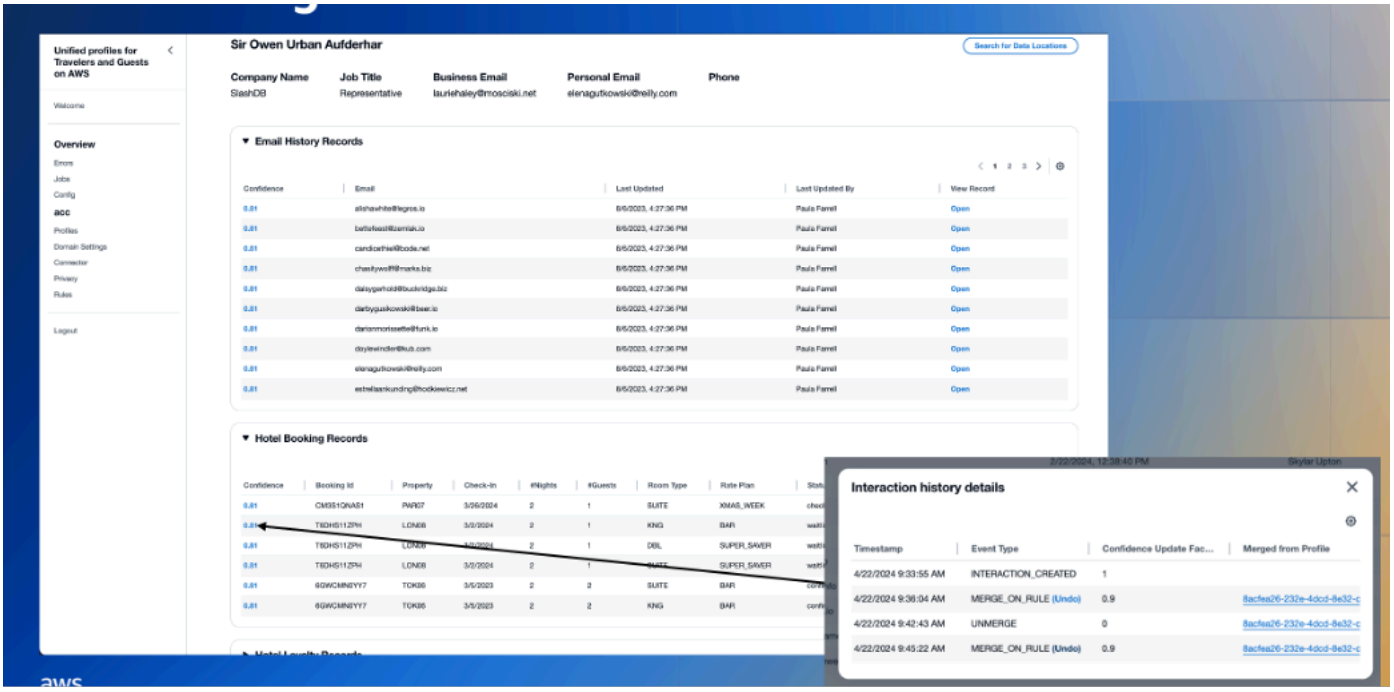
> **Note**
>
> The search criteria must be an exact match.

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 allows you to view all the data available for this profile along with the profile change history and data lineage.

A Confidence factor is provided for every interaction within a profile based on how the interaction reached the profile (from native ingestion, Rule based, manual or AI based matching).



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:

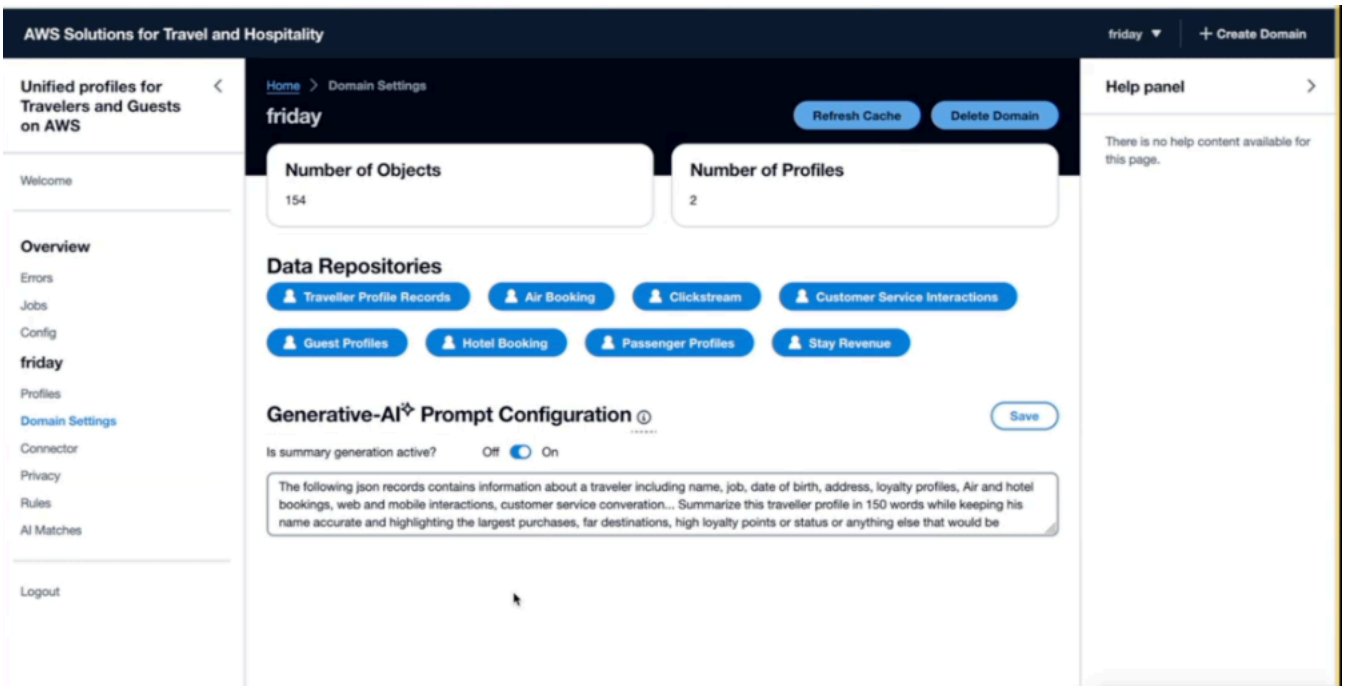
```
https://central-res.com/{{ hotelCode}}/{{ bookingId}}
```

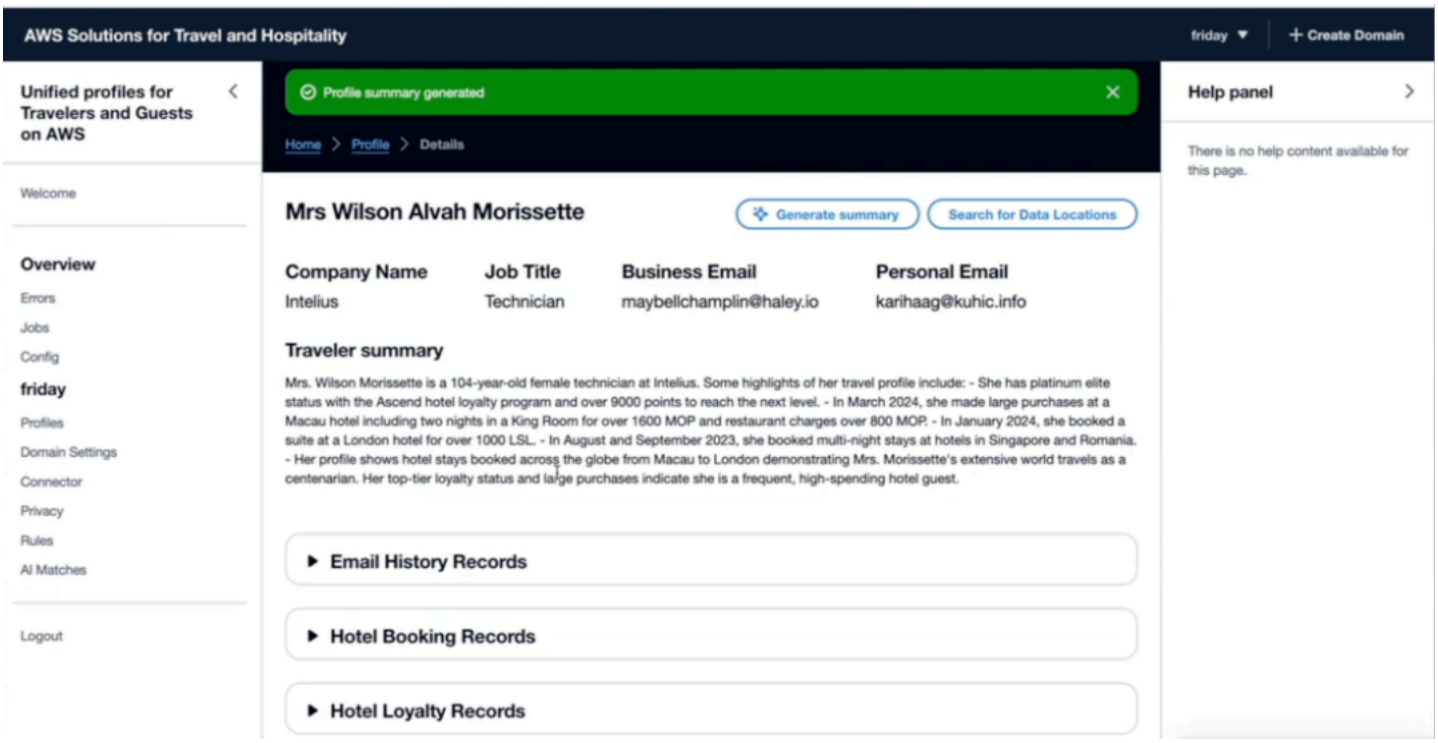
will render https://central-res.com/ABCDEF/12345679 and allow to user to navigate to that URL.

Enable Generative AI Summarization

In version 2.0.0 and above, the profile display screen can optionally display a summary of the profile content created using Amazon Bedrock (AWS Generative AI Service). Follow the steps below to enable this capability.

1. Make sure access to Anthropic Claude 3.0 Sonnet is enabled in the AWS Account by following the steps described in the Amazon Bedrock documentation.
2. Go to the solution user interface, access the domain configuration page, and enable the Generative AI Summarization feature by selecting the **toggle UI element**.
3. Once enabled, you can choose to leave the default prompt or customize the prompt to meet your specific business objective (see below how the solution constructs the final prompt by joining the configured prompt with the profile data).
4. Once the prompt is set, the changes take effect in real time. Display any traveler profile then choose the **Generate summary** button.





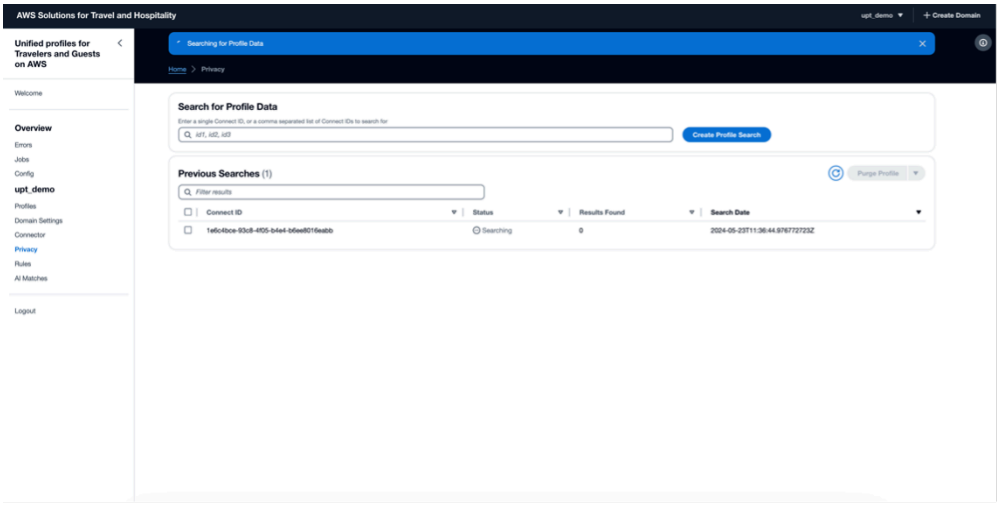
Privacy search and delete

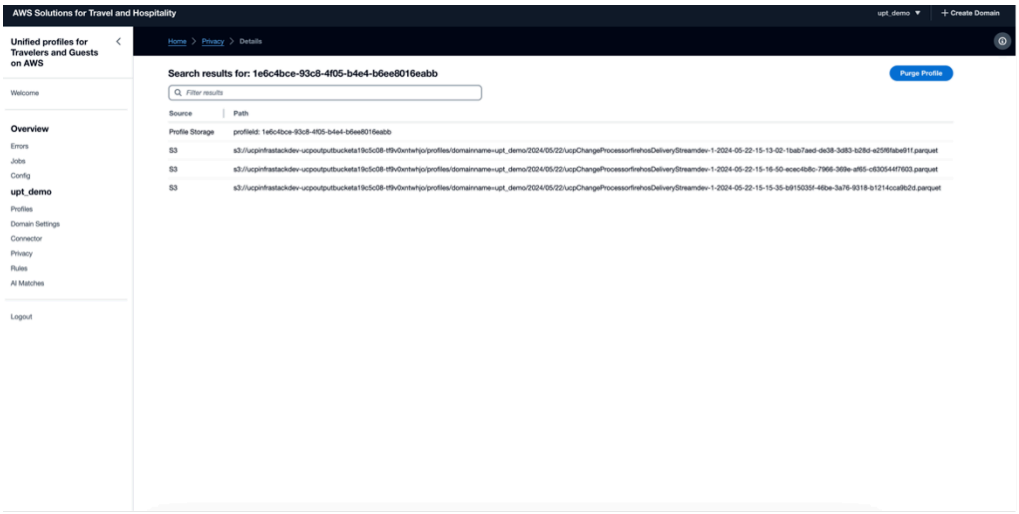
From version 2.0.0 on, the solution provides a dedicated screen allowing customers to batch search and delete profiles in order to facilitate compliance with privacy laws like GDPR. You can access this capability from the solution front end by selecting on the Privacy menu item. The search panel allows you to search batch and individual profile IDs.

Once the search completes, you can choose to view all storage locations where data has been found for a given profile and delete the data.

Note that the deletion process is irreversible (even history is cleared to ensure GDPR compliance). The ID of the Cognito user performing the delete operation is logged for compliance purpose.

Refer to the permission section to setup the necessary permission for users to be able to delete profile data.





Backup and restore the solution data

The solution's core design principle is that all non-recomputable data is stored in Amazon Aurora Serverless v2. This means that backing up the solution only requires backing up the Aurora cluster (and none of the other storage components like DynamoDB or Amazon Connect Customer Profiles). The solution offers a cache repopulation feature allowing you to recreate the data in the downstream caches (DynamoDB, Amazon Connect Customer Profiles) following an Aurora DB restore.

To backup and restore your Amazon Aurora cluster, follow the instruction in the *Overview of backing up and restoring an Aurora DB cluster* in the Amazon Aurora documentation.

To repopulate the solution's downstream cache, follow the instruction below:

1. Log into the solution administration portal.

2. Go to your domain configuration page.
3. Select **Recompute caches.**

This last action initiates the execution of multiple Fargate tasks in parallel allowing fast recreation of cache data. Note that this operation can occur while processing live data without impact on the resulting profile data accuracy.

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.

Note that the API targets the Amazon Aurora interaction store directly and not the downstream caches (DynamoDB or Amazon Connect Customer Profiles). This means that the API is not suitable for large programmatic traffic (like data planes). The API is meant to be used for integration with a user interface.

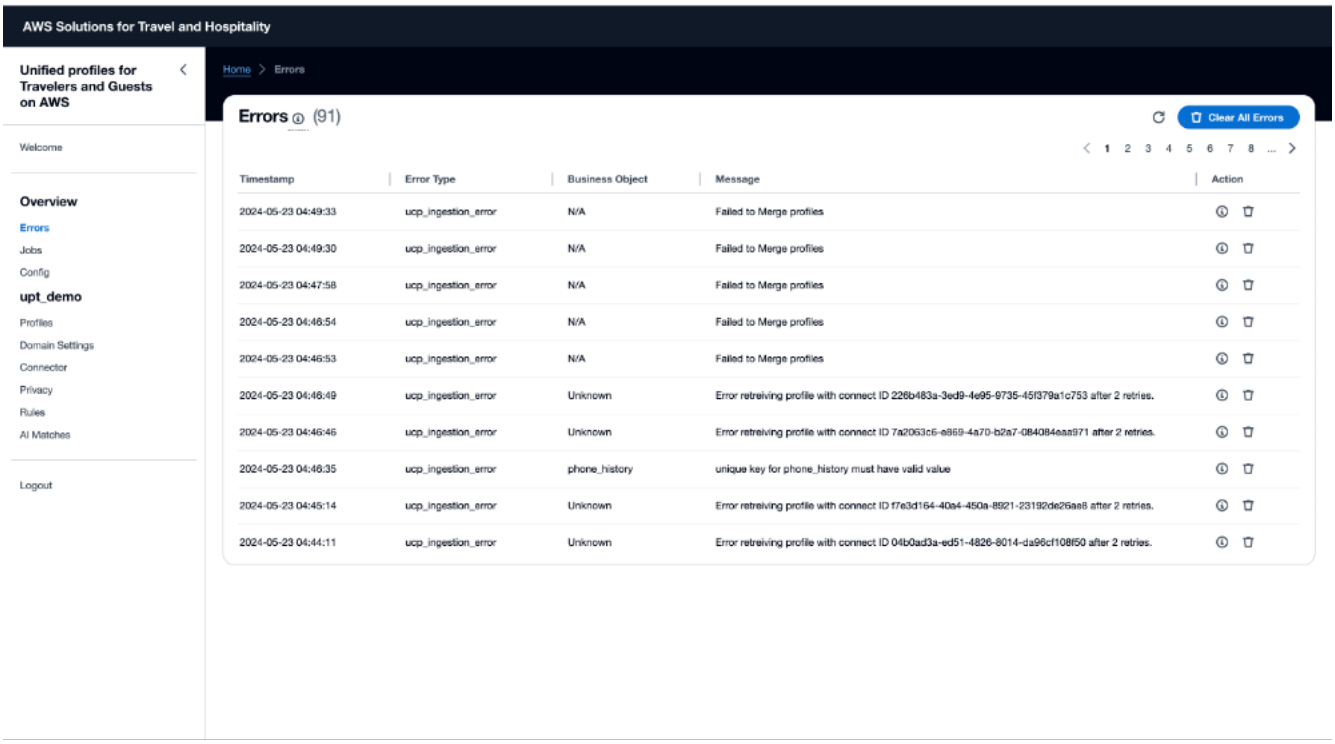
To integrate the search and retrieve profile within your application for processing large volumes of transactions, we recommend using the AWS API for Amazon Connect Customer Profiles and DynamoDB.

Monitoring the solution

The solution provides two main monitoring views.

The Errors page

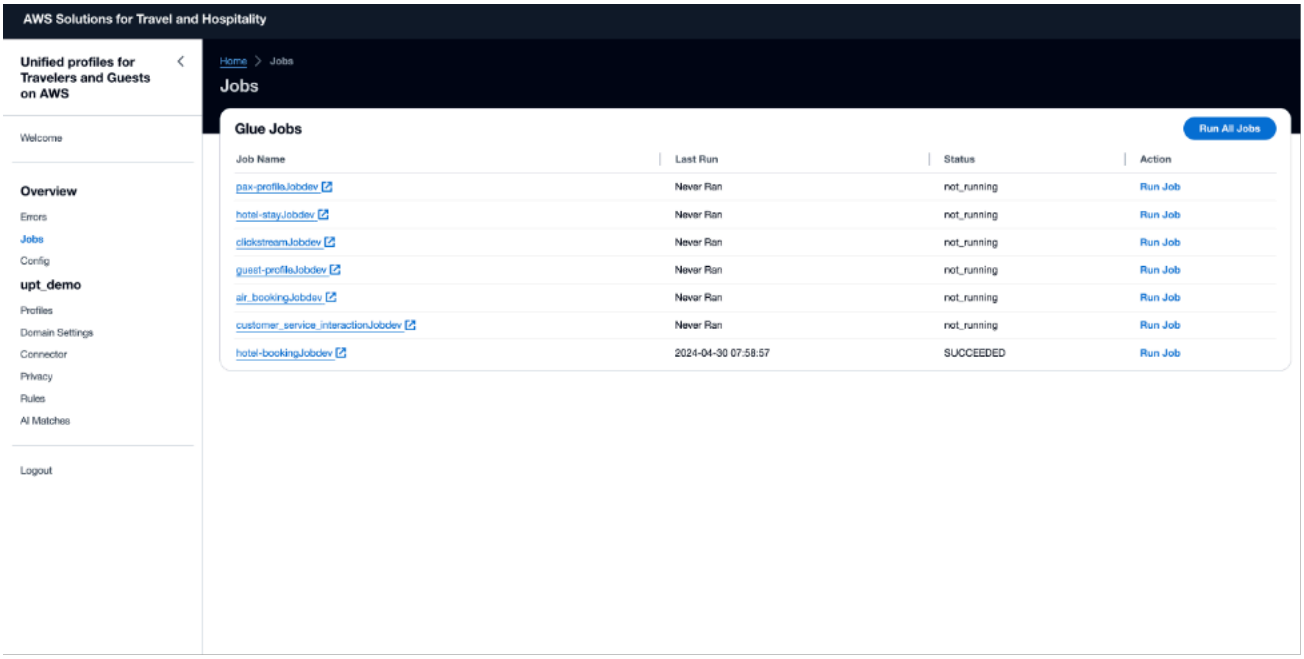
This page is located in the frontend deployed by the solution. This page lists all ingestion errors and the status of Glue jobs and Appflow runs. It also provides high-level metrics on the profile count and convenient access to the source data buckets. Below are a couple of examples:



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

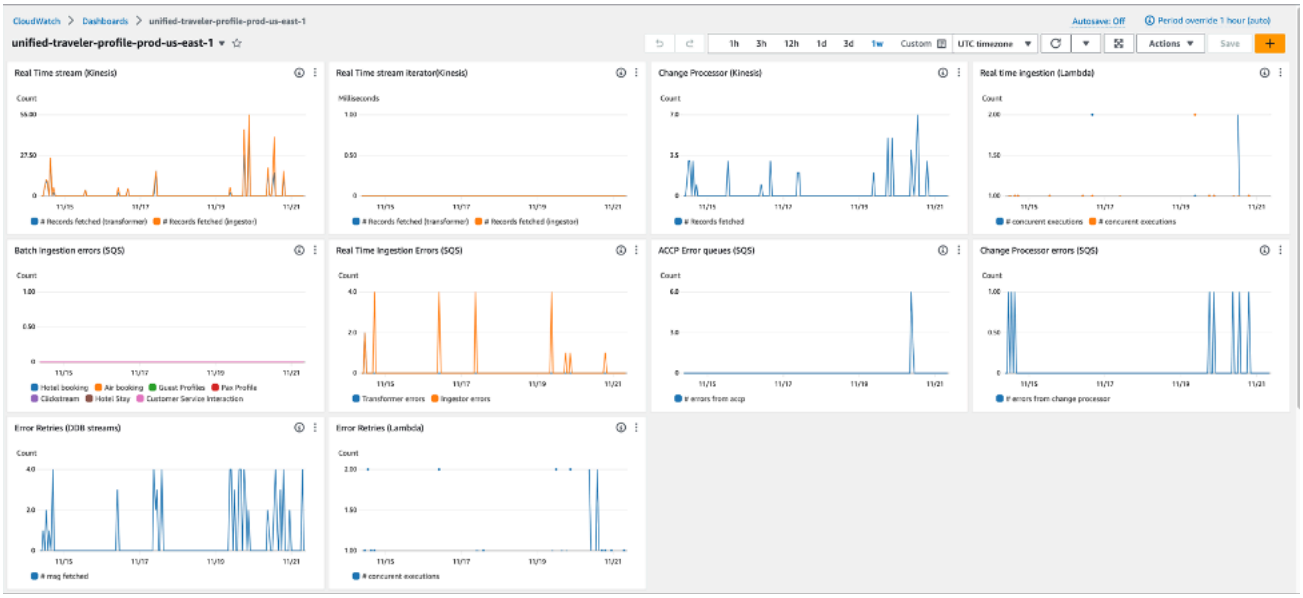
The Jobs page

This page provides a list of all batch ingestion jobs, their status and the ability to initiate the job run.



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

CloudWatch custom metrics

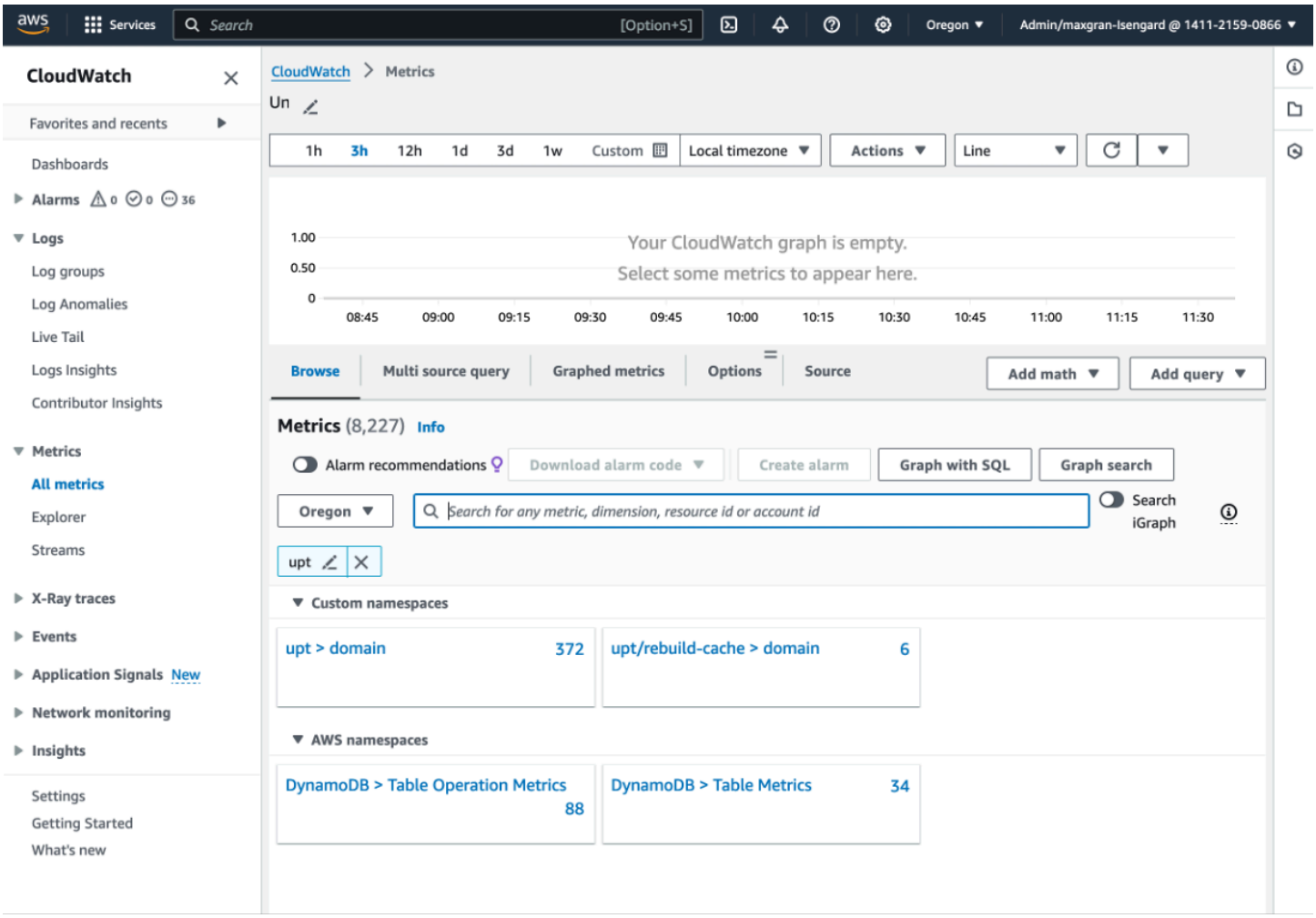
The solution creates the following list of custom metrics using CloudFront Embedded Metric Filters:

Metric name	Metric description
`PutProfileObjectRqCount`	PutProfileObject: Count of incoming request
`PutProfileObjectSuccessCount`	PutProfileObject: Count of success
`PrepProfileLatency`	PutProfileObject: Latency measurement for control plane access and profile preparation before insert
`InsertProfileLatency`	PutProfileObject: Latency measurement for insertion of profile in Amazon Aurora

Metric name	Metric description
UpdateDownstreamCache	PutProfileObject: Latency measurement for insertion of profile in downstream cache (dynamo/CP)
PutProfileObjectLatency	PutProfileObject: Insertion of profile in downstream cache (dynamo/CP)
DynamoUpdateLatency	PutProfileObject: Latency measurement for DynamoDB cache update
CPUpdateLatency	PutProfileObject: Latency measurement for CP Cache update
MergeProfileLatency	PutProfileObject: Latency measurement for CP Cache update
UnmergeProfileLatency	UnMergeProfile: Latency measurement for overall profile unmerge
PutProfileRuleBasedLatency	Latency measurement for rule-based matching execution
IndexProfileLatency	Latency measurement for update of the match table
RuleExecutionCount	Count of rule execution by rule ID and version
AuroraInsertRollbackCount	Count of error inserting object into Aurora
DynamoInsertErrorCount	Count of error inserting object into DynamoDB
CPInsertErrorCount	Count of error inserting object into CP

These metrics are designed to provide detailed monitoring of the sub components of the interaction store to identify potential bottlenecks specific to your workload. This can provide insights to AWS Support while troubleshooting performance issues.

To visualize these metrics, go to the Amazon CloudWatch console, go to Metrics, and choose the name space "upt" as illustrated below.



CloudWatch console filtered to upt metrics

Amazon CloudWatch Logs

The solution creates logs in Amazon CloudWatch with a default 10 year retention period.

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.

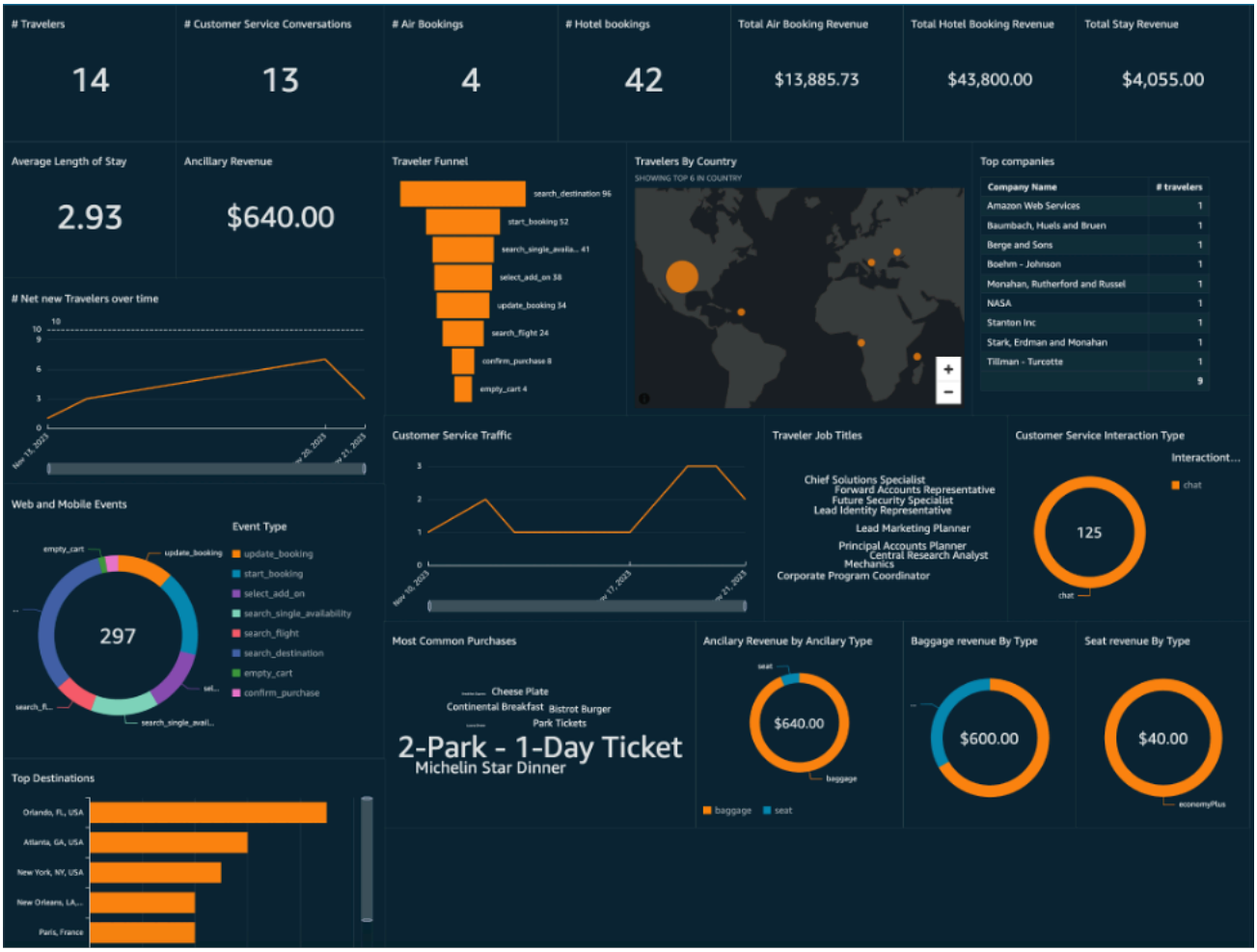
> **Note**
>
> The solution sends the full traveler profile for every batch of changes occurring withing the same few seconds. This allows bus subscribers to use the full profile data to engage the traveler in real time using a downstream system.

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>

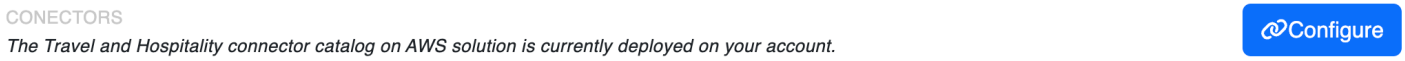
Any additional processes that produces matches with the same schema in DynamoDB will natively be supported by the solution. Customers can use AWS partners like Amperity or their own batch fuzzy-matching logic to augment the solution's capability accordingly.

> **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.



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

1. Download the `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

- Akash Garg
- Ahern Knox
- Brad Hong
- Beomseok Lee
- Geoffroy Rollat
- Hardik Khare
- Max Granat

- Owen Brady
- Ryan Love
- 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.
August 2024	Release v2.0.0: Non-backward-compatible change. Introduction of the interaction store built on Amazon Aurora to support data lineage and profile unmerge. Addition of Privacy search, Rule-based stitching at interaction level. New front-end design with a console-like look-and-feel.
August 2024	Release v2.0.1: Ensure all AWS resources are properly tagged with customer-provided tags. Upgrade to dependencies and minor bug fixes.

Date	Change
September 2024	Documentation update: Included instructi ons for setting up AI based Identity Resolutio n in the Deploy the solution section of the implementation guide.
October 2024	Release v2.0.2: Updated package versions to resolve security vulnerabilities.

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.