Implementation Guide

Travel and Hospitality Application Connectors Catalog on AWS

Travel and Hospitality Application Connectors Catalog on AWS: Implementation Guide

Table of Contents

Connect with key travel and hospitality data

Publication date: *August 2023 (last update: December 2023)*

The Travel and Hospitality Application Connectors Catalog on AWS solution helps you connect with key systems in the travel and hospitality industry and establish data feeds. It provides access to industry data otherwise locked inside core systems, such as hotel property management systems (PMS), airline passenger service systems (PSS) or loyalty program software systems. These connectors allow you to move your industry data to AWS in real time, monitor and analyze your data using foundational analytics features, and seamlessly integrate with other AWS travel and hospitality solutions.

This solution establishes connectivity to key property management systems through AWS Partner HapiCloud, the airline and hotel brand website, and mobile application clickstream using AWS Partner Tealium EventStream.

The solution provides three stream connectors: Amazon Data Firehose, Amazon Kinesis Data Streams, and Amazon API Gateway. These connectors allow a source system to push data to AWS services.

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

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

If you want to . . .	Read . . .
Know the cost for running this solution.	Cost
Understand the security considerations for this solution.	Security
Know which AWS Regions are supported for this solution.	Supported AWS Regions
View or download the AWS CloudForm ation template included in this solution	AWS CloudFormation template

If you want to . . .	Read . . .
to automatically deploy the infrastructure resources (the "stack") for this solution.	

The guide is intended for IT architects, developers, DevOps, data analysts, and marketing technology professionals who want to implement the solution in their environment.

Features and benefits

The solution provides the following features and benefits:

Accelerate your data lake construction project on AWS

With this solution, AWS takes on the work of ingesting your Property Management System (PMS) and website traffic data to AWS, saving you time.

Create an analytics dashboard to extract value out of your property data and website data in minutes

All data is ingested and transformed in a common format that is accessible as an Amazon Athena data table. This allows AWS to create an Amazon QuickSight dashboard on top of the data in minutes.

Improve your guest experience with a real-time event-driven response to your PMS events

Every PMS event initiates an Amazon S3 PUT event that you can subscribe to in Amazon EventBridge.

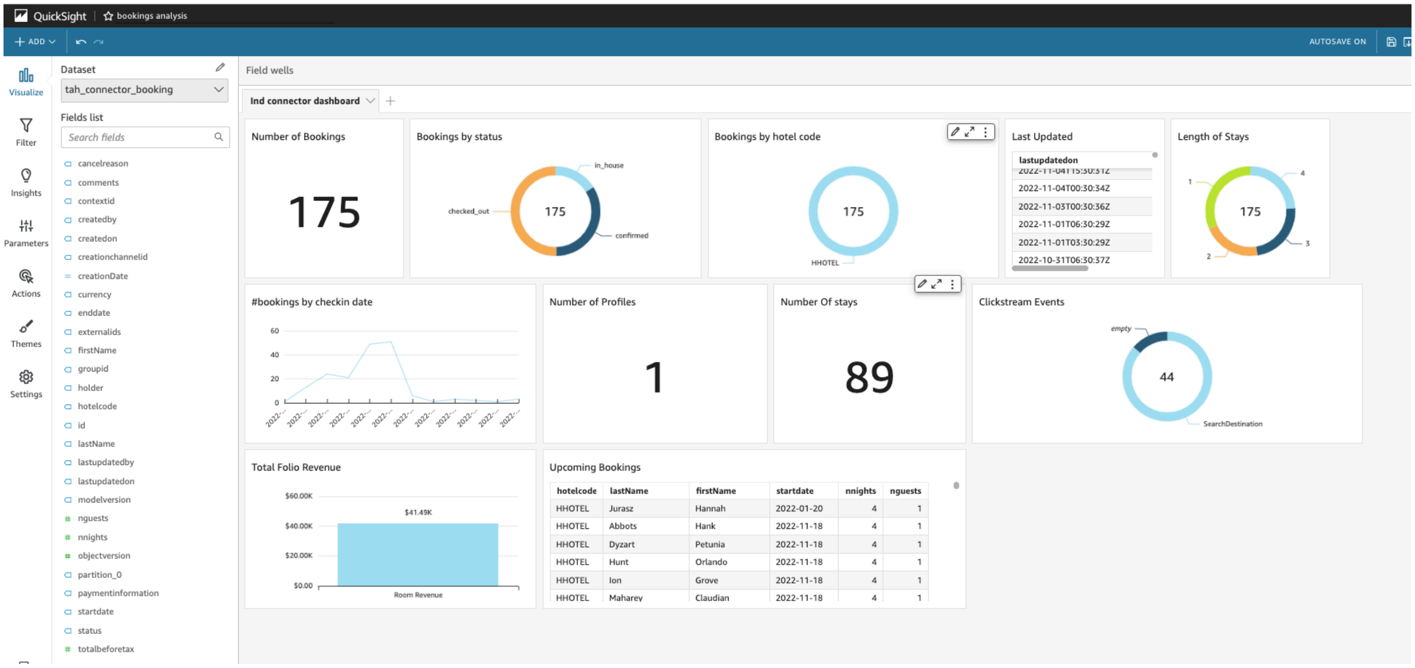
Build a forward compatible data repository to interact with future AWS Travel and Hospitality solutions

The solution allows you to transform incoming PMS and website traffic data into a standard AWS supported format used by AWS and AWS partners.

Use cases

- Ingestion of Opera, Infor, and other PMS data into a data lake
- Ingestion of Tealium travel website traffic data into a data lake

- Unified traveler profile: build a unified view of your guests and travelers
- Low-code analytics platform for PMS and website data



Last Updated

lastupdatedon
2022-11-04T15:30:31Z
2022-11-04T00:30:34Z
2022-11-03T00:30:36Z
2022-11-01T06:30:29Z
2022-11-01T03:30:29Z
2022-10-31T06:30:37Z

Upcoming Bookings

hotelcode	lastName	firstName	startdate	nnights	nguests
HHOTEL	Jurasz	Hannah	2022-01-20	4	1
HHOTEL	Abbots	Hank	2022-11-18	4	1
HHOTEL	Dyzart	Petunia	2022-11-18	4	1
HHOTEL	Hunt	Orlando	2022-11-18	4	1
HHOTEL	Ion	Grove	2022-11-18	4	1
HHOTEL	Maharey	Claudian	2022-11-18	4	1

Amazon QuickSight dashboard with Athena property and website data

Concepts and definitions

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

Connector type

The solution has one type of connector:

- Streaming connectors integrate with external systems and Amazon services to receive events from their source systems either through an AWS Streaming service (like Amazon Kinesis), a web hook using Amazon API Gateway, or an IoT broker.

AWS Travel and Hospitality data model

All data is ingested and transformed into an AWS Travel and Hospitality data model, which is the standard for AWS Travel and Hospitality solutions.

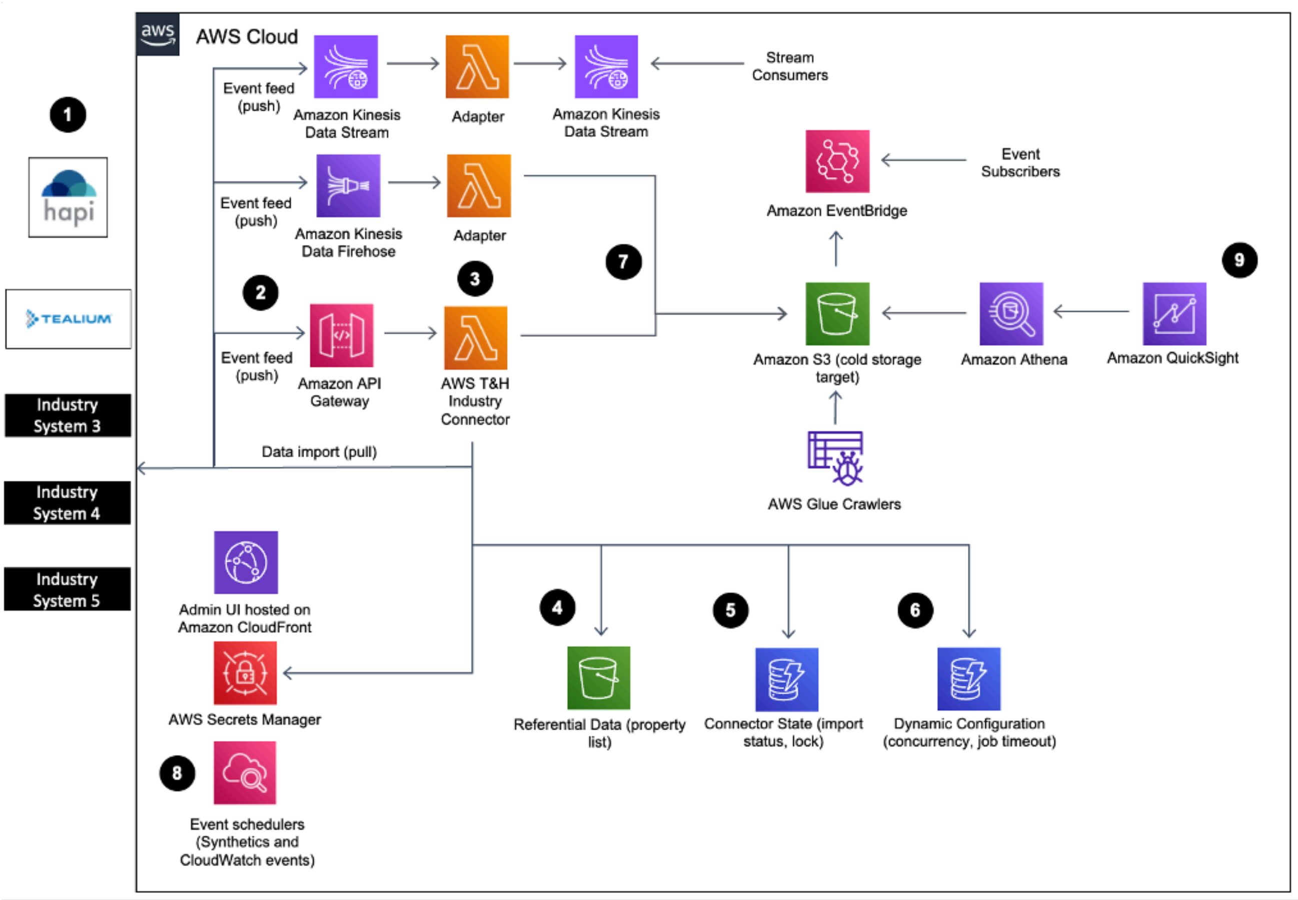
For a general reference of AWS terms, refer to the AWS glossary in the AWS General Reference.

Architecture overview

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

Architecture diagram

Deploying this solution with the default parameters builds the following environment in your AWS account.


AWS Cloud
1
hapi
TEALIUM
Industry System 3
Industry System 4
Industry System 5
Event feed (push)
Amazon Kinesis Data Stream
Adapter
Amazon Kinesis Data Stream
Stream Consumers
Event feed (push)
Amazon Kinesis Data Firehose
Adapter
Amazon EventBridge
Event Subscribers
2
3
7
9
Event feed (push)
Amazon API Gateway
AWS T&H Industry Connector
Amazon S3 (cold storage target)
Amazon Athena
Amazon QuickSight
Data import (pull)
AWS Glue Crawlers
Admin UI hosted on Amazon CloudFront
AWS Secrets Manager
4
5
6
Referential Data (property list)
Connector State (import status, lock)
Dynamic Configuration (concurrency, job timeout)
8
Event schedulers (Synthetics and CloudWatch events)

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

1. The solution launches with HapiCloud PMS connector and Tealium integration. Third-party system credentials are managed in AWS Secret Manager.

2. The connector offers a PUSH mode. PUSH allows the source system to publish the data change to AWS through REST API (webhook) or by writing data to Amazon Data Firehose and Amazon Kinesis Data Streams.
3. The connector logic manages the imports and publish process, converts the data into the AWS industry model, and ensures the data reaches the target service. The backend functionalities are organized in use cases (search reservation, import reservation, receive clickstream event, etc.) and are entirely serverless.
4. The solution includes Athena as a storage space allowing customers to import relevant industry data needed for the data import, such as list of properties, currency, and code mappings.
5. The solution uses Amazon DynamoDB to maintain state and manage concurrency when needed. This allows the connector to scale up to process billions of records while providing users with real-time status monitoring capabilities.
6. The solution uses DynamoDB to store connector configuration (such as field mappings) when needed.
7. The solution ultimately sends the data to various AWS services (depending on the connector type) such as Amazon Simple Storage Service (Amazon S3) for cold storage, Amazon Kinesis for data streaming, or Amazon EventBridge for real-time data. At this point, the data is converted into the AWS industry model.
8. Amazon CloudWatch offers a convenient way for a solution user to schedule and initiate connector use-case initiation based on their need.
9. The solution comes with QuickSight that provides visibility into the import process and high-level metrics on the business data.

AWS Well-Architected design considerations

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

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

Operational excellence

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

This AWS solution pushes metrics to Amazon CloudWatch at various stages to provide observability into the infrastructure; Lambda functions, Amazon S3 buckets, and the rest of the solution components. Continuous integration and continuous delivery (CI/CD) and infrastructure deployment are managed in code through AWS CDK and AWS CodePipeline. The HapiCloud platform allows on-demand and automated data replay to prevent data loss. The Tealium connector uses the Amazon Data Firehose processor, which logs incorrectly processed records in an Amazon S3 subfolder.

Security

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

- The solution user interface access and the Hapi ingestion layer are authorized with Amazon Cognito using two separate AppClients.
- The Tealium ingestion layer is protected by IAM.
- All inter-service communications use AWS IAM roles.
- All roles used by the solution follows least-privilege access. That is, it only contains minimum permissions required so the service can function properly.
- Communication to end users and Amazon API Gateway uses a Bearer token generated and handled by Amazon Cognito.
- Data secured at rest and in transit are important security considerations that cut through AWS specific language.
- Hapi connector token API access is restricted to a specific IP range provided by HapiCloud.
- All code in this solution has been reviewed using an extensive list of automated tools (including `cdk-nag` and `cfn-nag`) and by the Amazon Security Engineering Group.

Reliability

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

- Using AWS Serverless Services wherever possible (for example, AWS Lambda, API Gateway, Amazon S3, and Kinesis) to ensure high availability and recovery from service failure.
- Using AWS Lambda functions for data processing. Data is stored in DynamoDB and Amazon S3, so it persists in multiple Availability Zones (AZs) by default.

Performance efficiency

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

- Using serverless architecture throughout this solution.
- The ability to launch this solution in any region that supports AWS services in this solution such as: AWS Lambda, Amazon API Gateway, and Amazon S3.
- Automatically testing and deploying this solution daily. Solution architects and subject matter experts reviewing this solution for areas to experiment and improve.
- The solution monitors the response time of the connectors by logging it to Amazon CloudWatch.

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 customers are charged based on use.

Sustainability

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

The solution utilizes managed and serverless services to minimize the environmental impact of the backend services. The solution, Serverless design, (using Lambda and DynamoDB) 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.

Backend software component

The solution provides a backend component allowing the user interface to configure connectors and retrieve connector status and a source system with a webhook interface to post data updates to the connector. The Lambda-hosted backend component follows a modular architecture allowing additional use cases and connectors to be created in a flexible and reliable way.

Firehose delivery stream

The solution deploys a Firehose delivery stream allowing clickstream data from travel and hospitality websites to be ingested at scale and formatted in the AWS data model. The data transformation leverages the Kinesis Firehose Data transformation feature and AWS Lambda. The user must provide data mapping for the event name and event attribute names using the solution administration portal.

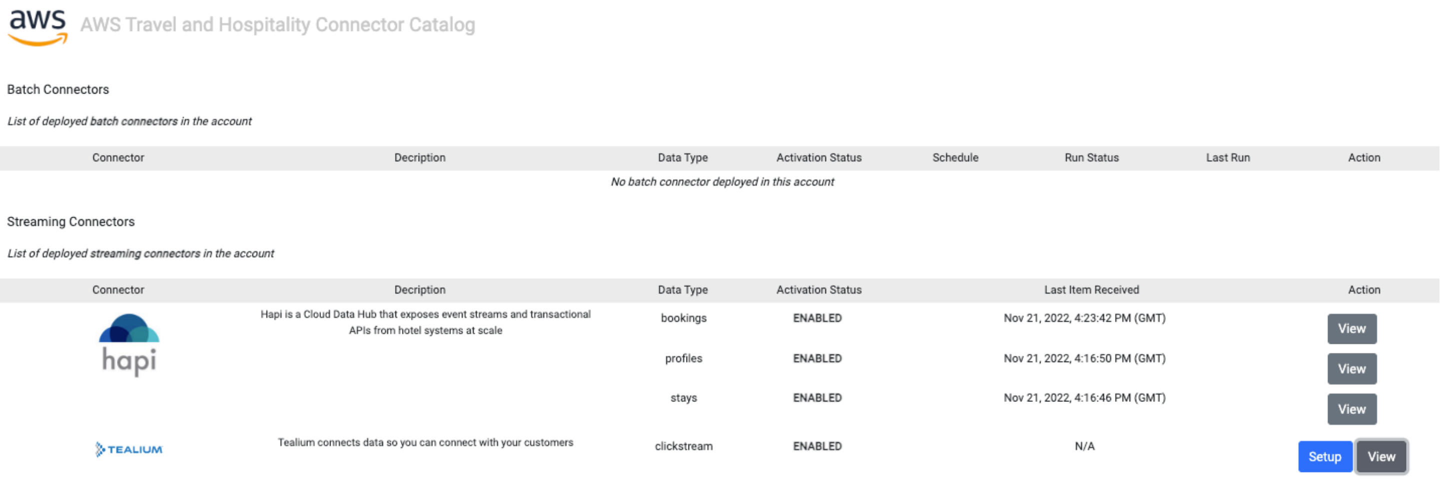
Components

This solution deploys the following components:

Administration portal

A user interface written in Angular allowing the solution administrator to configure and monitor the activation of the stream connectors

The website is hosted in Amazon CloudFront. The Amazon Cognito console manages user access.



Batch Connectors

List of deployed batch connectors in the account

Connector	Decription	Data Type	Activation Status	Schedule	Run Status	Last Run	Action
No batch connector deployed in this account							

Streaming Connectors

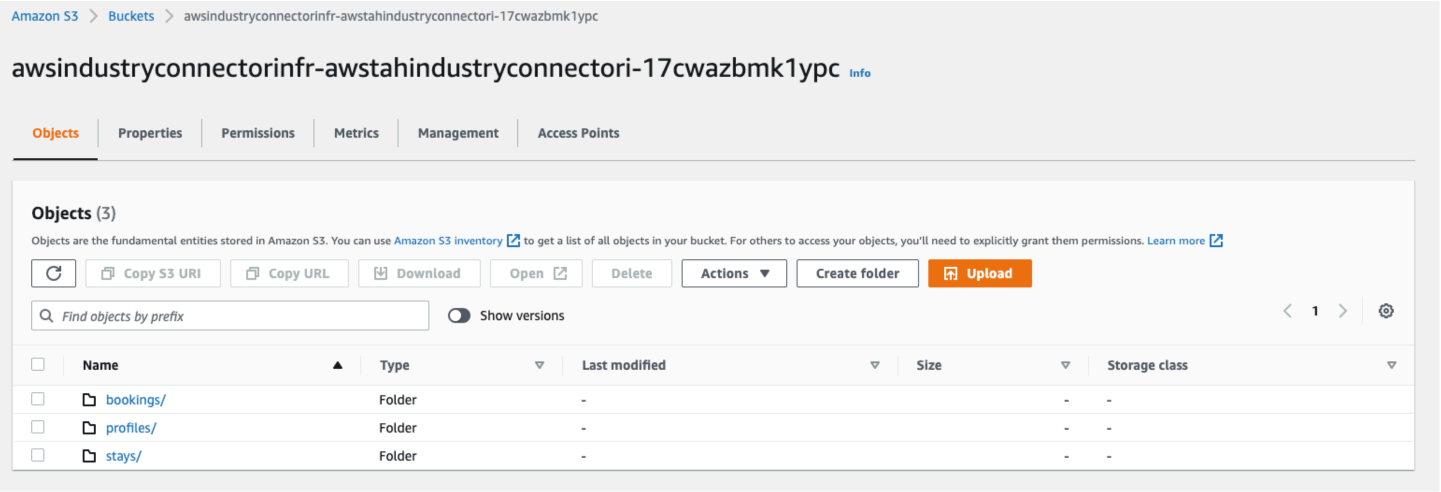
List of deployed streaming connectors in the account

Connector	Decription	Data Type	Activation Status	Last Item Received	Action
hapi	Hapi is a Cloud Data Hub that exposes event streams and transactional APIs from hotel systems at scale	bookings	ENABLED	Nov 21, 2022, 4:23:42 PM (GMT)	View
		profiles	ENABLED	Nov 21, 2022, 4:16:50 PM (GMT)	View
		stays	ENABLED	Nov 21, 2022, 4:16:46 PM (GMT)	View
TEALIUM	Tealium connects data so you can connect with your customers	clickstream	ENABLED	N/A	Setup View

Travel and Hospitality Application Connectors Catalog solution administrator portal

Data storage connector analytics component

The primary target of data import use cases is Amazon S3. This follows AWS data lake best practices and allows you to further transform this data for import in downstream AWS services like Amazon Redshift. A unique bucket is created for all data import. This allows easy access management for the solution users. Each object type is imported under its own folder.



awsindustryconnectorinfr-awstahindustryconnectori-17cwazbmk1ypc

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

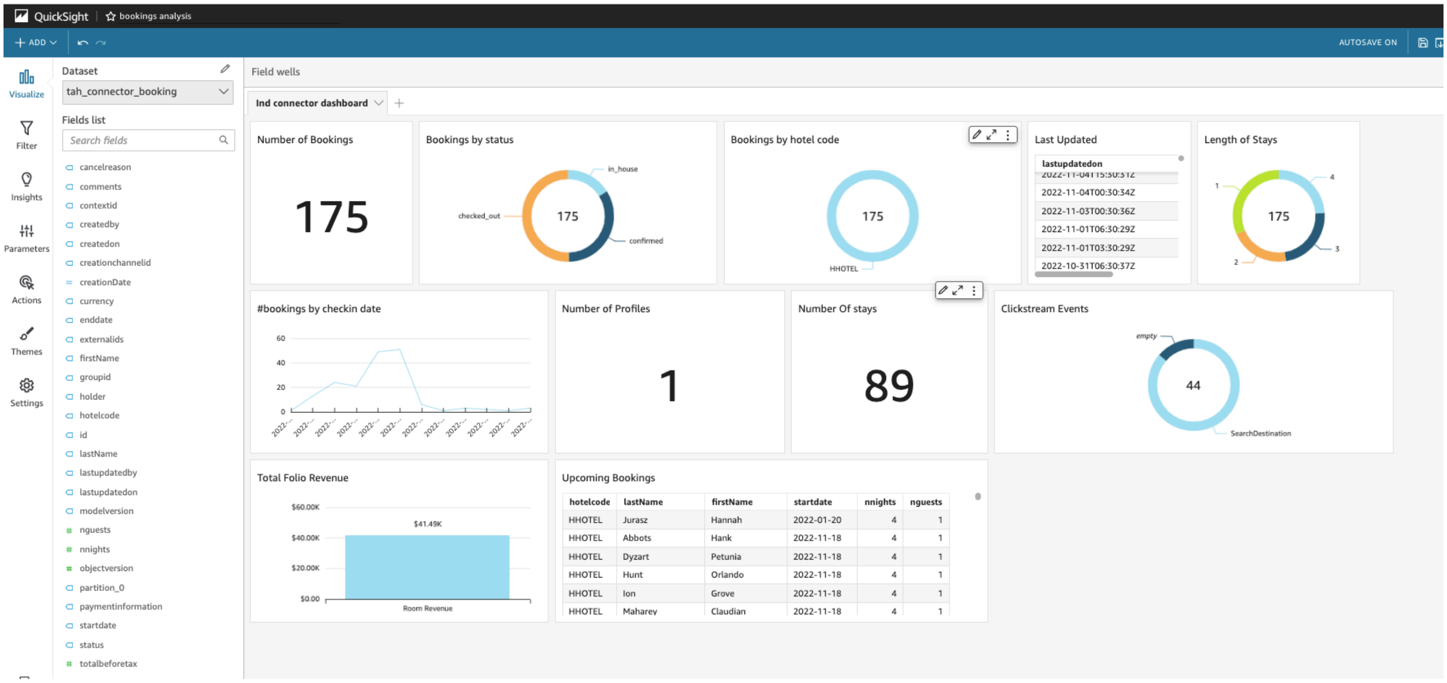
□	Name	Type	Last modified	Size	Storage class
□	bookings/	Folder	-	-	-
□	profiles/	Folder	-	-	-
□	stays/	Folder	-	-	-

Data storage connector analytics component folders

The solution creates an AWS Glue crawler for each S3 path which generates an Amazon Athena table that you can use to analyze the imported data with tools such as Amazon QuickSight. This capability provides a turnkey analytics platform for small and medium customers on top of their industry data, which allows more advanced customers to monitor the import process.

Note

Only the top-level element of the business objects will be usable for analytics. Customers looking for more advanced analytics use cases can use [AWS Glue Studio](#) to flatten the data records.



Last Updated

lastupdatedon
2022-11-04T15:30:31Z
2022-11-04T00:30:34Z
2022-11-03T00:30:36Z
2022-11-01T06:30:29Z
2022-11-01T03:30:29Z
2022-10-31T06:30:37Z

Upcoming Bookings

hotelcode	lastName	firstName	startdate	nnights	nguests
HHOTEL	Jurasz	Hannah	2022-01-20	4	1
HHOTEL	Abbots	Hank	2022-11-18	4	1
HHOTEL	Dyzart	Petunia	2022-11-18	4	1
HHOTEL	Hunt	Orlando	2022-11-18	4	1
HHOTEL	Ion	Grove	2022-11-18	4	1
HHOTEL	Maharey	Claudian	2022-11-18	4	1

QuickSight Dashboard

AWS services in this solution

AWS service	Description
[AWS CloudFormation](#)	**Core**. This solution uses [AWS CloudFormation templates and stacks](#) to automate its deployment.
[AWS Lambda](#)	**Core**. This solution uses AWS Lambda to transfer the incoming traffic into Business Objects compatible with the AWS travel and hospitality schema.

AWS service	Description
Amazon Data Firehose	**Supporting**. The solution uses Kinesis Data Firehose to receive and process the incoming clickstream data from Tealium and store it in Amazon S3.
Amazon DynamoDB	**Supporting**. The solution uses DynamoDB to store the ingestion status of the business objects.
Amazon S3	**Supporting**. The solution uses Amazon S3 as a main storage system for the ingested data. Amazon S3 also generates `PutObject` events allowing customers to setup real-time event streams.
Amazon API Gateway	**Supporting**. The solution creates an API Gateway HTTP endpoint for each data type (supported data types in version 1.0 are booking, profiles and stays). The solution receives these data and transform them into an AWS travel and hospitality data model.
Amazon Cognito	**Supporting**. The solution uses Amazon Cognito to secure access to its frontend.
Amazon Athena	**Supporting**. The solution uses Amazon Athena to explore and analyze the content of the ingested data.
AWS Glue	**Supporting**. Customers can use AWS Glue to crawl the ingested data and the data in Amazon S3 content to setup a downstream data lake.
Amazon CloudFront	**Supporting**. The solution uses Amazon CloudFront to deploy its frontend.

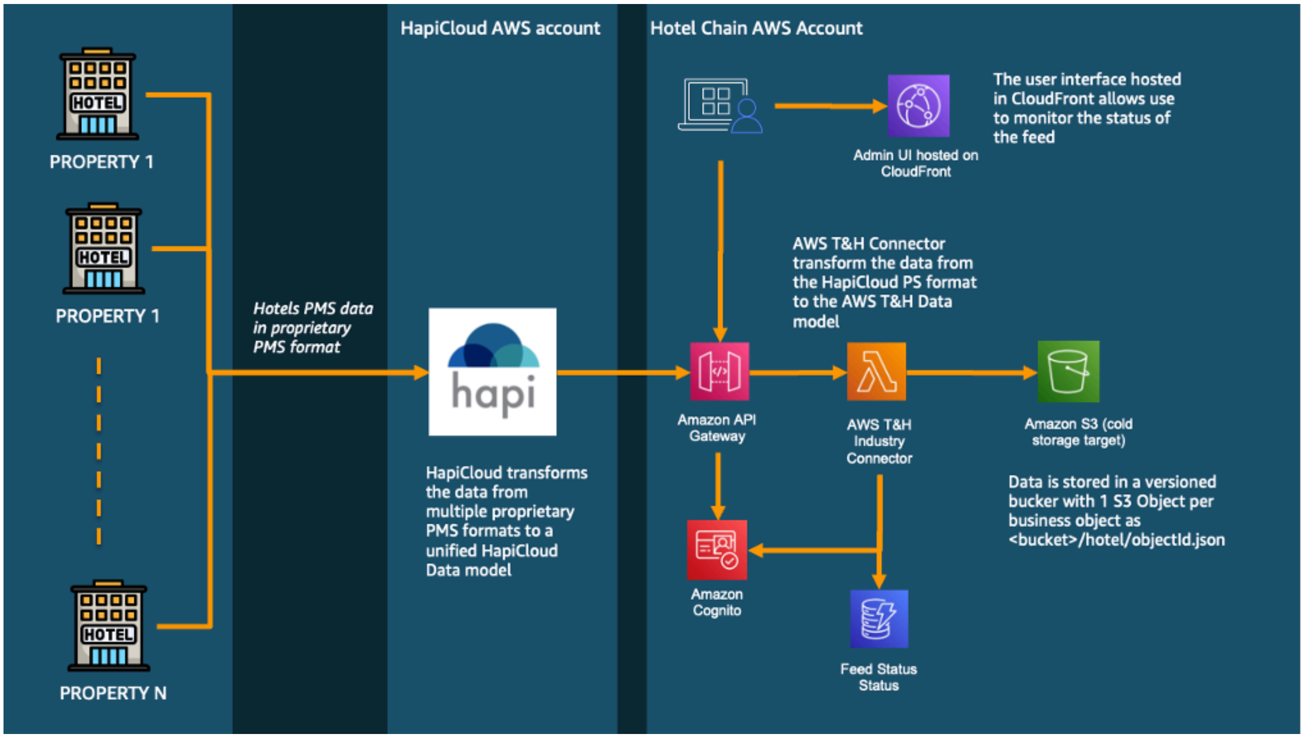
AWS service	Description
Amazon QuickSight	**Optional**. The solution can use Amazon QuickSight to explore and visualize the content of the ingested data.

How the solution works

Third-party connector details

HapiCloud PMS connector

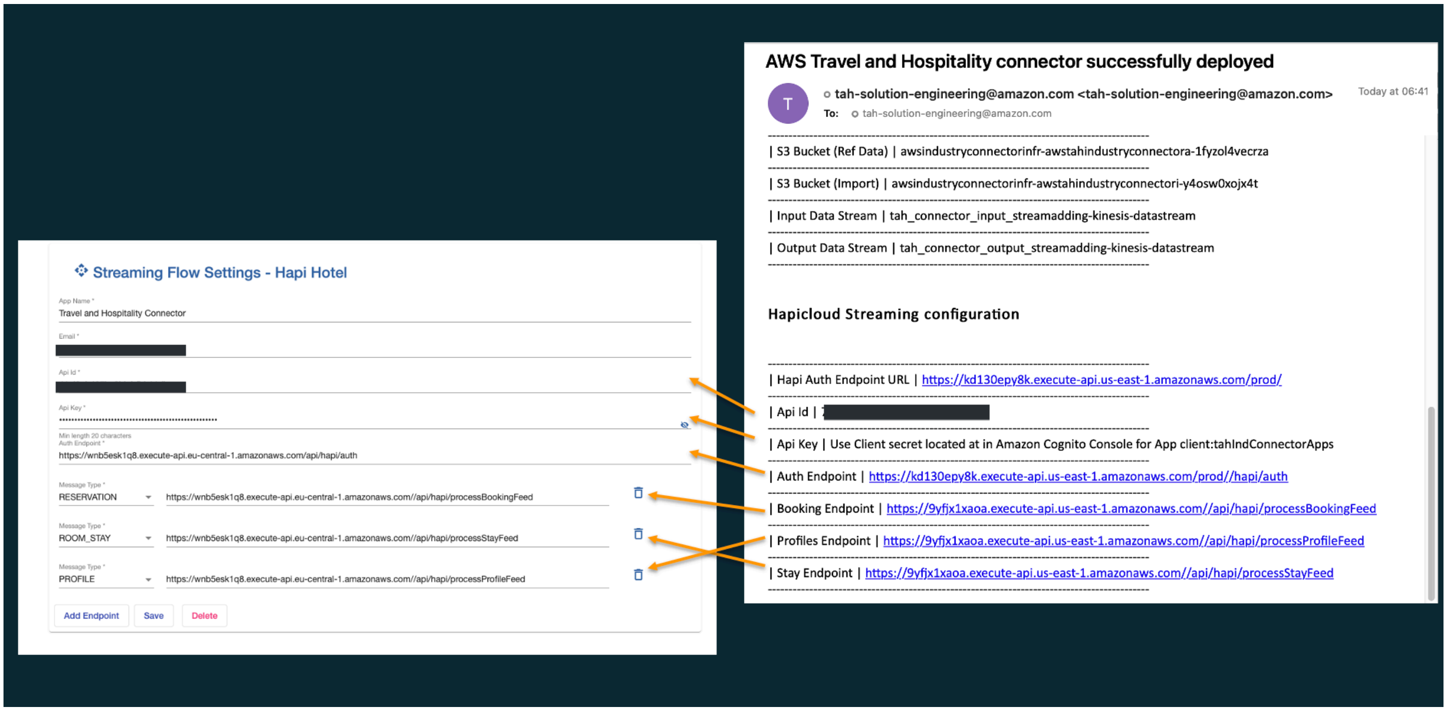
The HapiCloud connector contains the following architecture:


HapiCloud AWS account
Hotel Chain AWS Account
PROPERTY 1
PROPERTY 1
PROPERTY N
Hotels PMS data in proprietary PMS format
hapi
HapiCloud transforms the data from multiple proprietary PMS formats to a unified HapiCloud Data model
Admin UI hosted on CloudFront
The user interface hosted in CloudFront allows use to monitor the status of the feed
AWS T&H Connector transform the data from the HapiCloud PS format to the AWS T&H Data model
Amazon API Gateway
AWS T&H Industry Connector
Amazon S3 (cold storage target)
Data is stored in a versioned bucker with 1 S3 Object per business object as <bucket>/hotel/objectId.json
Amazon Cognito
Feed Status Status

The HapiCloud PMS connector receives real-time updates for booking, stays, and profiles from the PMS supported by Hapi. Note that historical data may optionally be resent by Hapi upon request. The connector is based on Hapi streaming flow capabilities, which allows Hapi users to set up a webhook allowing Hapi to post real-time changes to hotel property data to a remote HTTP endpoint.

The solution creates an API Gateway HTTP endpoint for each data type (booking, profiles, and stays). The solution receives these data and transforms them into an AWS travel and hospitality data model.

To set up the connector, input the API endpoints and credentials provided by the CloudFormation script output (and received by email if you added an email address in input of the script).



Streaming Flow Settings - Hapi Hotel

App Name *
Travel and Hospitality Connector

Email *
[redacted]

Api Id *
[redacted]

Api Key *
••
Min length 20 characters

Auth Endpoint *
https://wnb5esk1q8.execute-api.eu-central-1.amazonaws.com/api/hapi/auth

Message Type *
RESERVATION https://wnb5esk1q8.execute-api.eu-central-1.amazonaws.com//api/hapi/processBookingFeed

Message Type *
ROOM_STAY https://wnb5esk1q8.execute-api.eu-central-1.amazonaws.com//api/hapi/processStayFeed

Message Type *
PROFILE https://wnb5esk1q8.execute-api.eu-central-1.amazonaws.com//api/hapi/processProfileFeed

AWS Travel and Hospitality connector successfully deployed

tah-solution-engineering@amazon.com <tah-solution-engineering@amazon.com> Today at 06:41
To: tah-solution-engineering@amazon.com

```
--------------------------------------------------------------------------------
| S3 Bucket (Ref Data) | awsindustryconnectorinfr-awstahindustryconnectora-1fyzol4vecrza
--------------------------------------------------------------------------------
| S3 Bucket (Import) | awsindustryconnectorinfr-awstahindustryconnectori-y4osw0xojx4t
--------------------------------------------------------------------------------
| Input Data Stream | tah_connector_input_streamadding-kinesis-datastream
--------------------------------------------------------------------------------
| Output Data Stream | tah_connector_output_streamadding-kinesis-datastream
--------------------------------------------------------------------------------
```

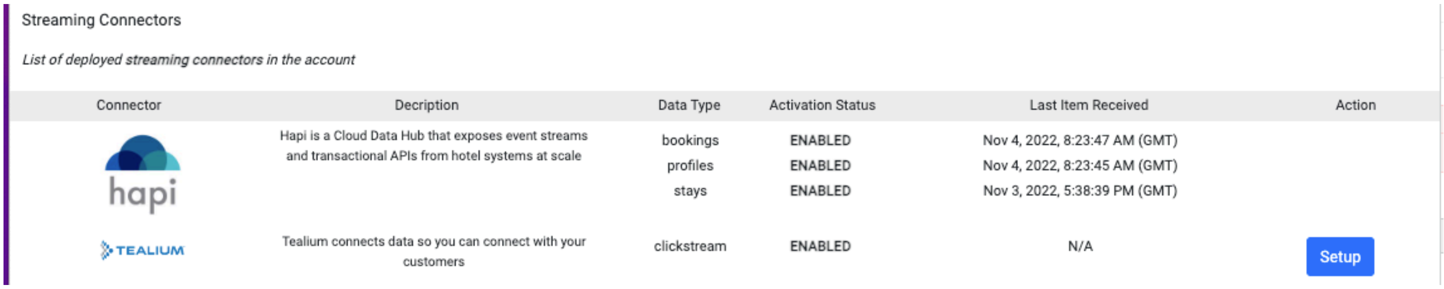
Hapicloud Streaming configuration

```
--------------------------------------------------------------------------------
| Hapi Auth Endpoint URL | https://kd130epy8k.execute-api.us-east-1.amazonaws.com/prod/
--------------------------------------------------------------------------------
| Api Id | [redacted]
--------------------------------------------------------------------------------
| Api Key | Use Client secret located at in Amazon Cognito Console for App client:tahIndConnectorApps
--------------------------------------------------------------------------------
| Auth Endpoint | https://kd130epy8k.execute-api.us-east-1.amazonaws.com/prod//hapi/auth
--------------------------------------------------------------------------------
| Booking Endpoint | https://9yfjx1xaoa.execute-api.us-east-1.amazonaws.com//api/hapi/processBookingFeed
--------------------------------------------------------------------------------
| Profiles Endpoint | https://9yfjx1xaoa.execute-api.us-east-1.amazonaws.com//api/hapi/processProfileFeed
--------------------------------------------------------------------------------
| Stay Endpoint | https://9yfjx1xaoa.execute-api.us-east-1.amazonaws.com//api/hapi/processStayFeed
--------------------------------------------------------------------------------
```

Providing for the CloudFormation script output

The API key, provided to Hapi, is the client secret generated by the Amazon Cognito app client created for programmatic access to the API. For details on how to obtain it from the AWS Management Console, see the Authentication section.

The Hapi administration portal validates both authentication and access to the stream API. To validate that data is being sent by Hapi, go to the AWS Solution administration portal and look at the Last Item Received column in the streaming connector status dashboard.



Streaming Connectors

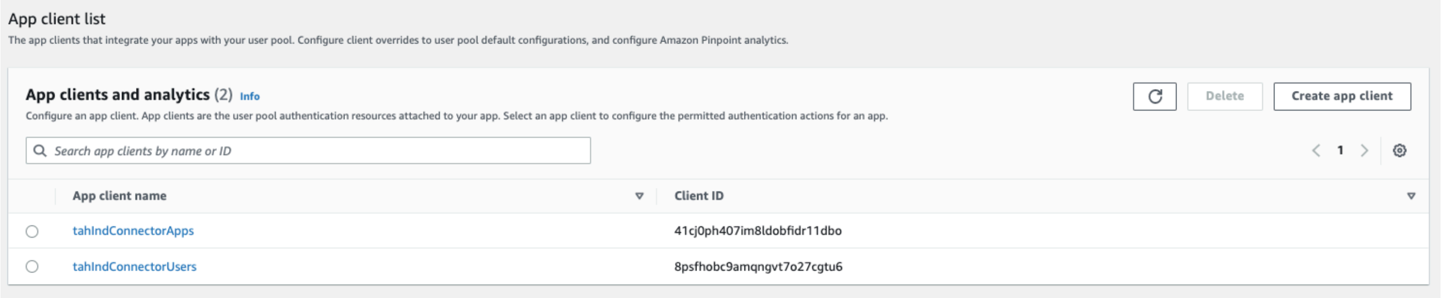
List of deployed streaming connectors in the account

Connector	Decription	Data Type	Activation Status	Last Item Received	Action
hapi	Hapi is a Cloud Data Hub that exposes event streams and transactional APIs from hotel systems at scale	bookings profiles stays	ENABLED ENABLED ENABLED	Nov 4, 2022, 8:23:47 AM (GMT) Nov 4, 2022, 8:23:45 AM (GMT) Nov 3, 2022, 5:38:39 PM (GMT)	
TEALIUM	Tealium connects data so you can connect with your customers	clickstream	ENABLED	N/A	Setup

Hapi administration portal

Authentication

The solution creates a Cognito user pool with two app integrations.



App client list

The app clients that integrate your apps with your user pool. Configure client overrides to user pool default configurations, and configure Amazon Pinpoint analytics.

App clients and analytics (2) Info

Configure an app client. App clients are the user pool authentication resources attached to your app. Select an app client to configure the permitted authentication actions for an app.

	App client name	Client ID
○	tahIndConnectorApps	41cj0ph407im8ldobfidr11dbo
○	tahIndConnectorUsers	8psfhobc9amqngvt7o27cgtu6

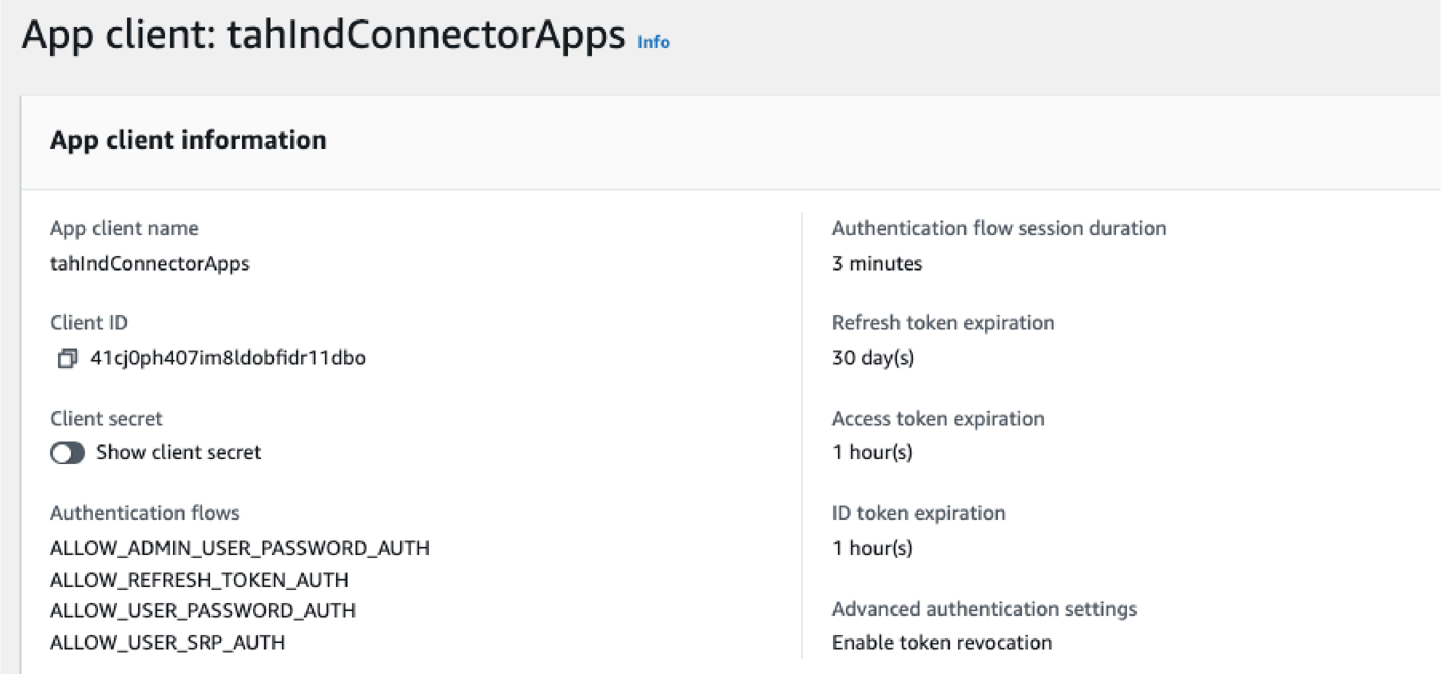
Cognito user pool with app integrations

The **tahIndConnectorApps** is used for programmatic access to the API using a client credential grant.

The **tahIndConnectorUsers** is used by the administration portal for user login.

To retrieve the client secret used by Hapi as an API key:

1. Sign in to the [Amazon Cognito console](https://console.aws.amazon.com/cognito).
2. Select the **tahIndConnectorApps** app client.
3. Select **Show client Secret** and copy the secret.



App client: tahIndConnectorApps Info

App client information

App client name
tahIndConnectorApps

Client ID
41cj0ph407im8ldobfidr11dbo

Client secret
Show client secret

Authentication flows
ALLOW_ADMIN_USER_PASSWORD_AUTH
ALLOW_REFRESH_TOKEN_AUTH
ALLOW_USER_PASSWORD_AUTH
ALLOW_USER_SRP_AUTH

Authentication flow session duration
3 minutes

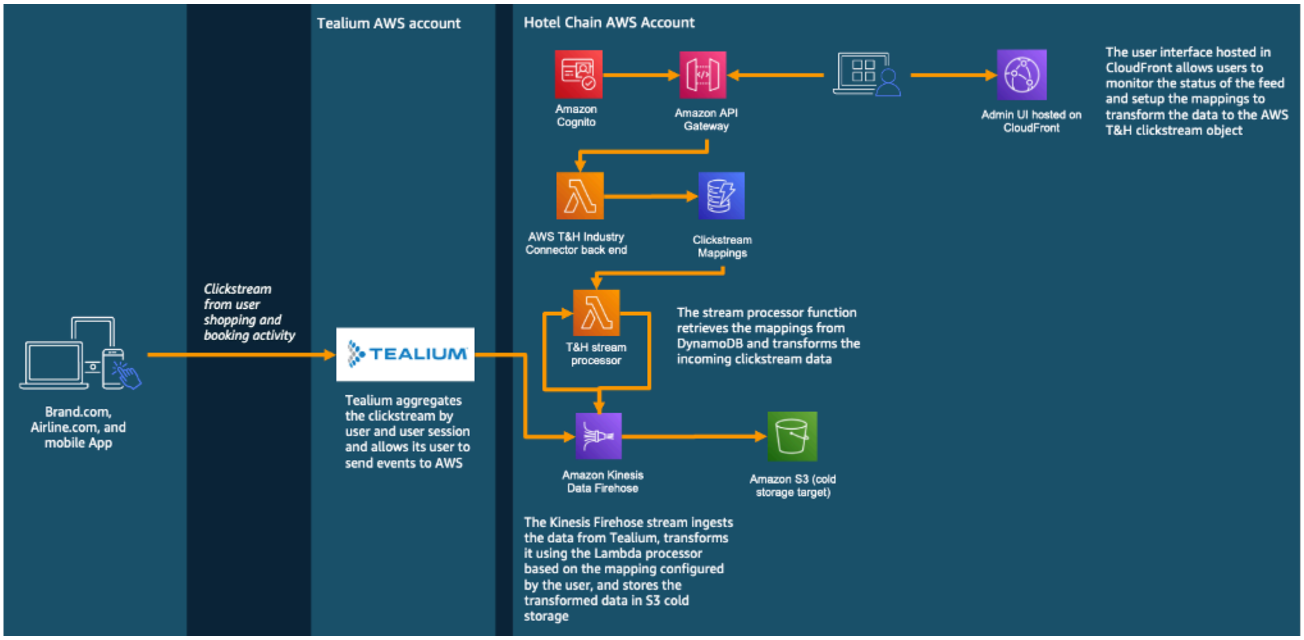
Refresh token expiration
30 day(s)

Access token expiration
1 hour(s)

ID token expiration
1 hour(s)

Advanced authentication settings
Enable token revocation

Tealium Event Stream connector


Tealium AWS account
Hotel Chain AWS Account
Amazon Cognito
Amazon API Gateway
Admin UI hosted on CloudFront
AWS T&H Industry Connector back end
Clickstream Mappings
T&H stream processor
Amazon Kinesis Data Firehose
Amazon S3 (cold storage target)
TEALIUM
Brand.com, Airline.com, and mobile App
Clickstream from user shopping and booking activity

The user interface hosted in CloudFront allows users to monitor the status of the feed and setup the mappings to transform the data to the AWS T&H clickstream object

The stream processor function retrieves the mappings from DynamoDB and transforms the incoming clickstream data

Tealium aggregates the clickstream by user and user session and allows its user to send events to AWS

The Kinesis Firehose stream ingests the data from Tealium, transforms it using the Lambda processor based on the mapping configured by the user, and stores the transformed data in S3 cold storage

Tealium offers a native integration with Amazon Data Firehose that allows the traveler clickstream to be ingested into Amazon S3 at scale. The solution allows you to map the clickstream data to the AWS data model using the Hapi administration portal.

Plan your deployment

This section describes the cost, security, Regions, 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 the revision, the cost for running this solution with the default settings in the **US East (N. Virginia)** is illustrated below depending on the connector type**.** These numbers are high-level estimates for AWS only and do not include the cost of HapiCloud or Tealium.

Connector	Dimensions	Cost [USD]
HapiCloud PMS Connector	per Year for one millions bookings per Year for one million stays per Year for one million profiles	$5.30 $5.30 $5.30
Tealium Audience Stream connector	per Year per Million site visits	$31.20
	Total:	**$47.10**

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

The cost of the solution can be monitored using the AWS Cost explorer or using AWS AppRegistry.

1. Open the AWS Management Console.
2. Go to AWS Systems Manager, select Application Manager, and `select AWS-AppRegistry: travel-and-hospitality-connector-*`.

Security

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

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda function read and write access to specific Amazon S3 buckets and Amazon DynamoDB tables created by the solution. The Lambda functions allow access to the Amazon Cognito user pool used to general temporary access token to secured the API Gateway endpoint.

Additionally, an IAM role is created to allow AWS Glue to crawl the imported data and make it available to Amazon Athena and QuickSight.

Supported AWS Regions

This AWS Solution has been validated in all Standard AWS Regions.

Region name	
US East (Ohio)	Asia Pacific (Seoul)
US East (N. Virginia)	Europe (Paris)
US West (Northern California)	Middle East (Bahrain)
US West (Oregon)	AWS GovCloud (US-West)
Africa (Cape Town)	

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

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

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

Time to deploy: Approximately 10 minutes

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: Post-configuration tasks

- Configure the source connector with output form the Pipeline.
- Create an Administration Portal user in Amazon Cognito.
- Set up the connectors from the Solution Administration portal.
- (Optional) Set up the Amazon QuickSight Dashboard.

⚠ **Important**

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

AWS CloudFormation template

Note

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

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

View template

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

Note

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

Prerequisites

You must complete the following prerequisite before launching the stack.

- Contact your HapiCloud or Tealium representative to get access to their platform according to your connector need.

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 10 minutes

1. Sign in to the AWS Management Console and select the button to launch the **tah-connector.yml** 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. Choose **Next**.
6. On the **Configure stack options** page keep the default options.
7. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
8. Choose **Create stack** to deploy the stack.

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

Step 2: Post-configuration tasks

Step 1: Configure the source connector with output form the Pipeline

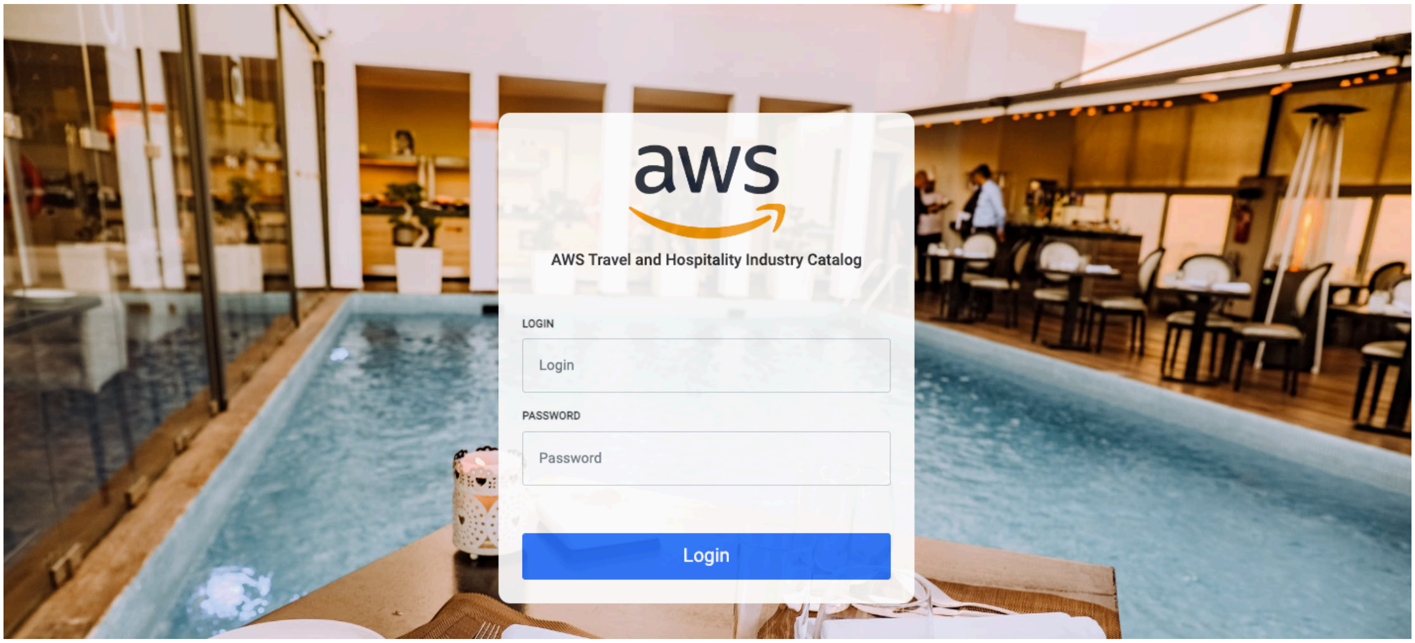
Once the deployment is complete, look for an email sent to the administrator email address provided in the CloudFormation template for the list of parameters required to configure the source connector of your choice (see the How the solution works section for detailed steps to configure each connector).

Step 2: Create an Administrator user using Amazon Cognito

Go to the Amazon Cognito console and follow the AWS documentation to create a user for the administration portal.

Step 3: Set up the connector from the Solution Administration portal

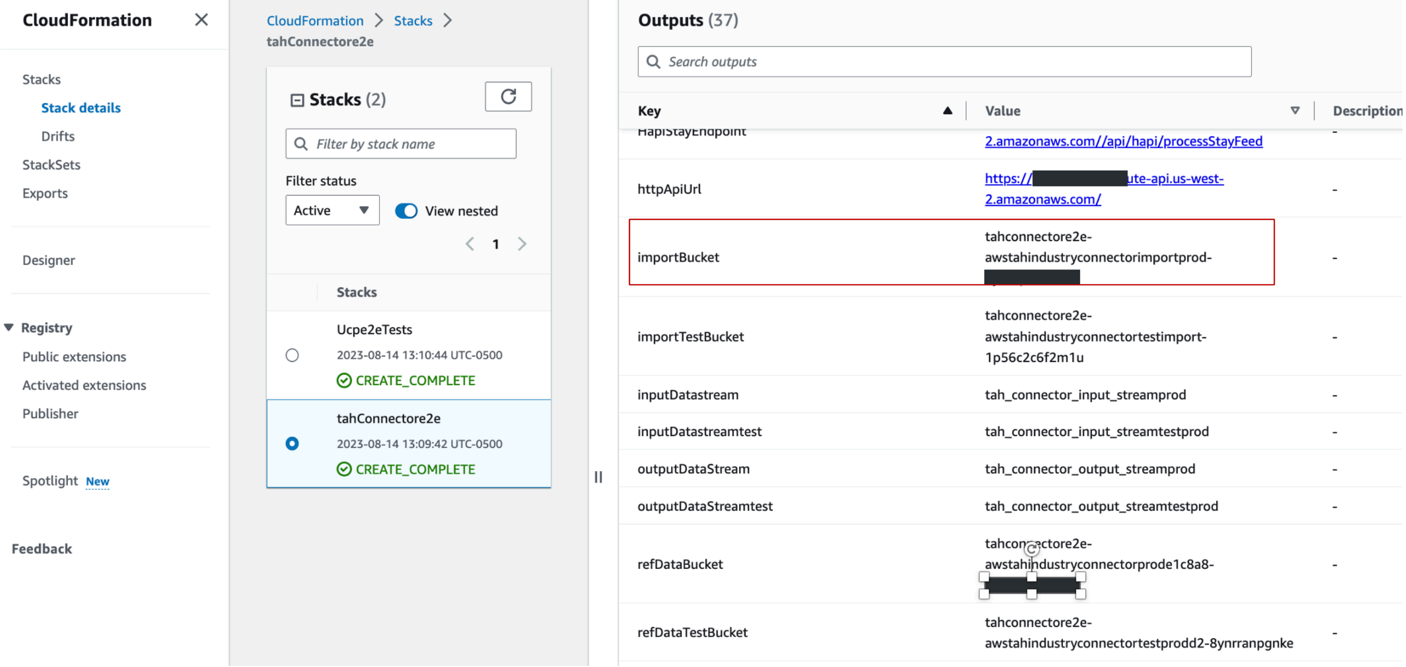
- Using the Amazon Cognito user, log in to the solution administration portal.



- Once logged in, refer to each dedicated section of this user guide to configure the connector appropriately (if needed).
- Use the administration portal to monitor that the connectors are active and have access to the Amazon S3 bucket where the data is stored. Requires AWS Management Console with the appropriate permissions.

[Optional] Set up the Amazon QuickSight Dashboard

1. Go to Amazon QuickSight and make sure your account is set up using the QuickSight Sign-in documentation.
2. On the top right menu, go to **Manage QuickSight.**
3. Go to **Security & Permission** to provide access to the Amazon S3 Bucket where the data is imported. (You must switch to `us-east-1` for this operation if you are in another Region.)
4. In the section **QuickSight access to AWS services**, select the Amazon S3 bucket where the data is stored (you can find the bucket name by selecting on the View button from the solution portal or in the output of CloudFormation under `importBucket`).

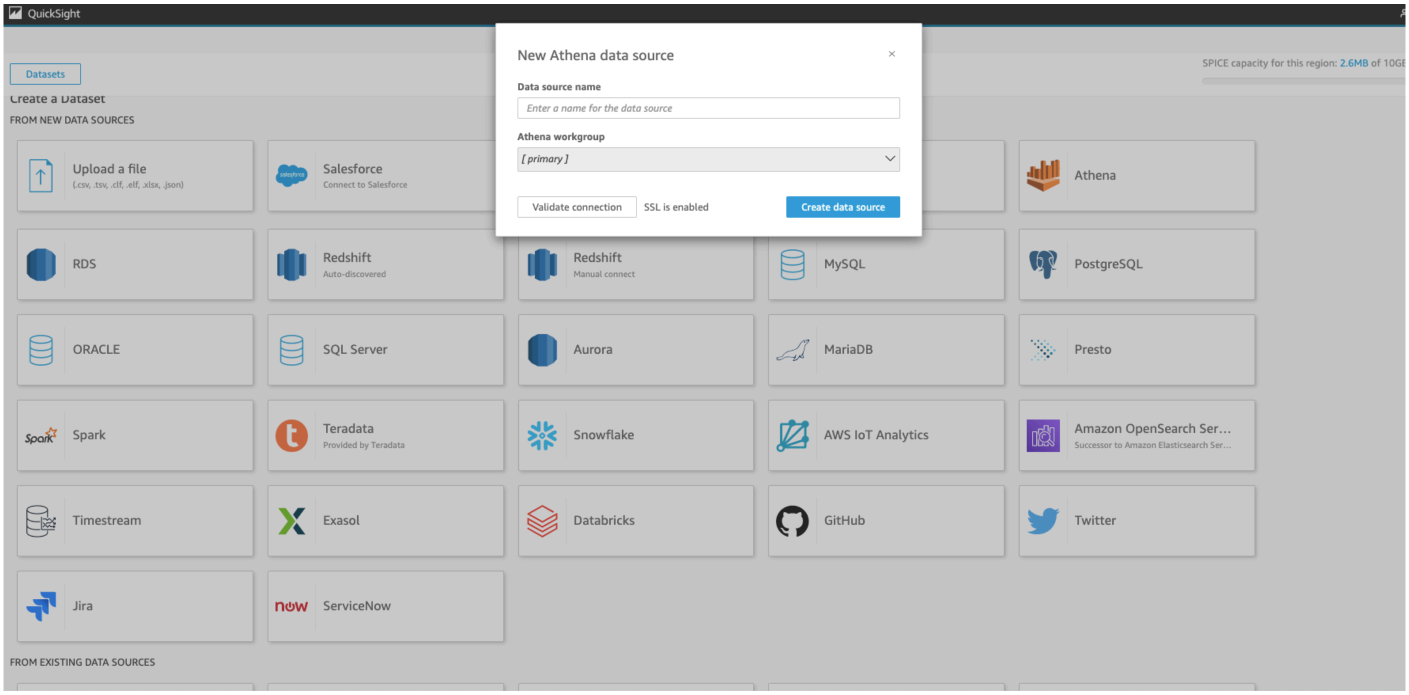


Stacks
○ Ucpe2eTests 2023-08-14 13:10:44 UTC-0500 CREATE_COMPLETE
● tahConnectore2e 2023-08-14 13:09:42 UTC-0500 CREATE_COMPLETE

Outputs (37)

Key	Value	Description
HapiStayEndpoint	2.amazonaws.com//api/hapi/processStayFeed	-
httpApiUrl	https://[redacted]ute-api.us-west-2.amazonaws.com/	-
importBucket	tahconnectore2e-awstahindustryconnectorimportprod-[redacted]	-
importTestBucket	tahconnectore2e-awstahindustryconnectortestimport-1p56c2c6f2m1u	-
inputDatastream	tah_connector_input_streamprod	-
inputDatastreamtest	tah_connector_input_streamtestprod	-
outputDataStream	tah_connector_output_streamprod	-
outputDataStreamtest	tah_connector_output_streamtestprod	-
refDataBucket	tahconnectore2e-awstahindustryconnectorprode1c8a8-[redacted]	-
refDataTestBucket	tahconnectore2e-awstahindustryconnectortestprodd2-8ynrranpgnke	-

5. Once access is provided, you can create a new dataset from a new Athena Data source.

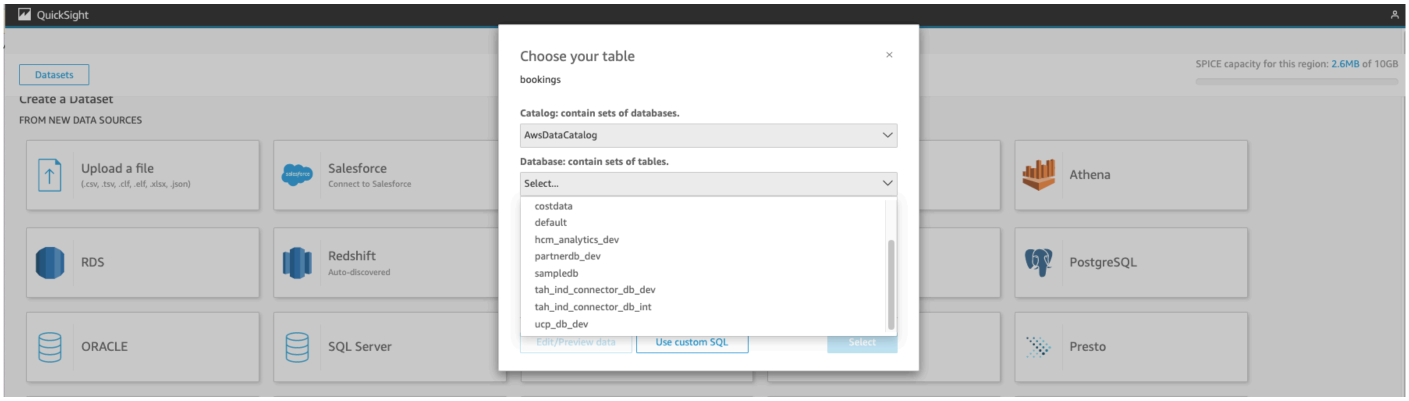


New Athena data source

Data source name: Enter a name for the data source

Athena workgroup: [primary]

6. Choose the tah_ind_connector_db_<env> database created by the solution.



Choose your table

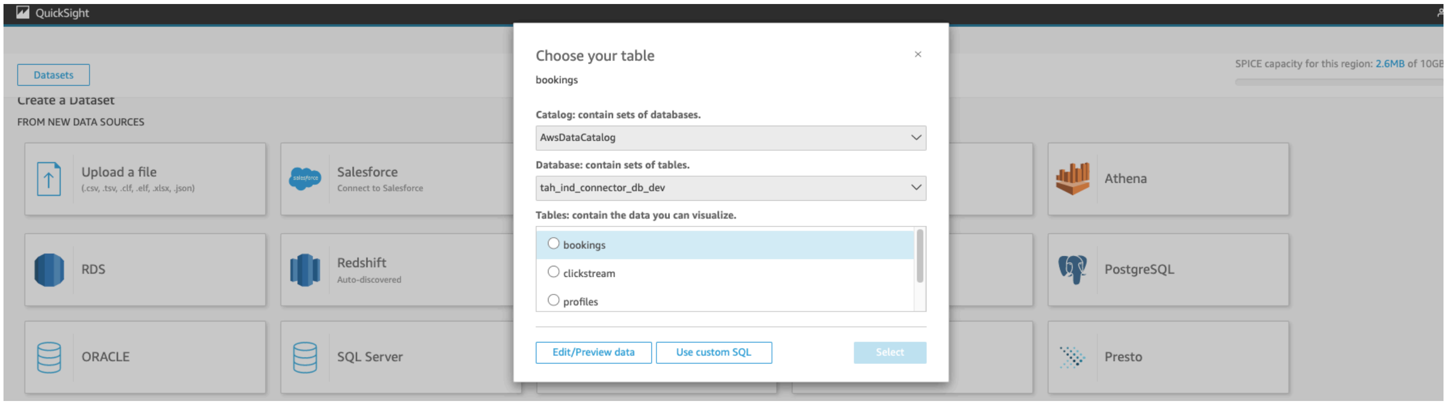
bookings

Catalog: contain sets of databases.
AwsDataCatalog

Database: contain sets of tables.
Select...

7. Choose the data type to visualize and select **Use custom SQL** to create a more advanced report.

> **Note**
>
> Custom SQL is used because the data stored in Amazon S3 is formatted in JSON with nested objects. Amazon Athena uses a function to access data such as first name, last name, rate plan, and prices.



Choose your table

bookings

Catalog: contain sets of databases.
AwsDataCatalog

Database: contain sets of tables.
tah_ind_connector_db_dev

Tables: contain the data you can visualize.
● bookings
○ clickstream
○ profiles

- The following three example of queries can help you create a dashboard:
  - In this example, we access the first holder first name and last name along with the amount before taxes of the first segment of a hotel booking segment and add these fields to the dataset

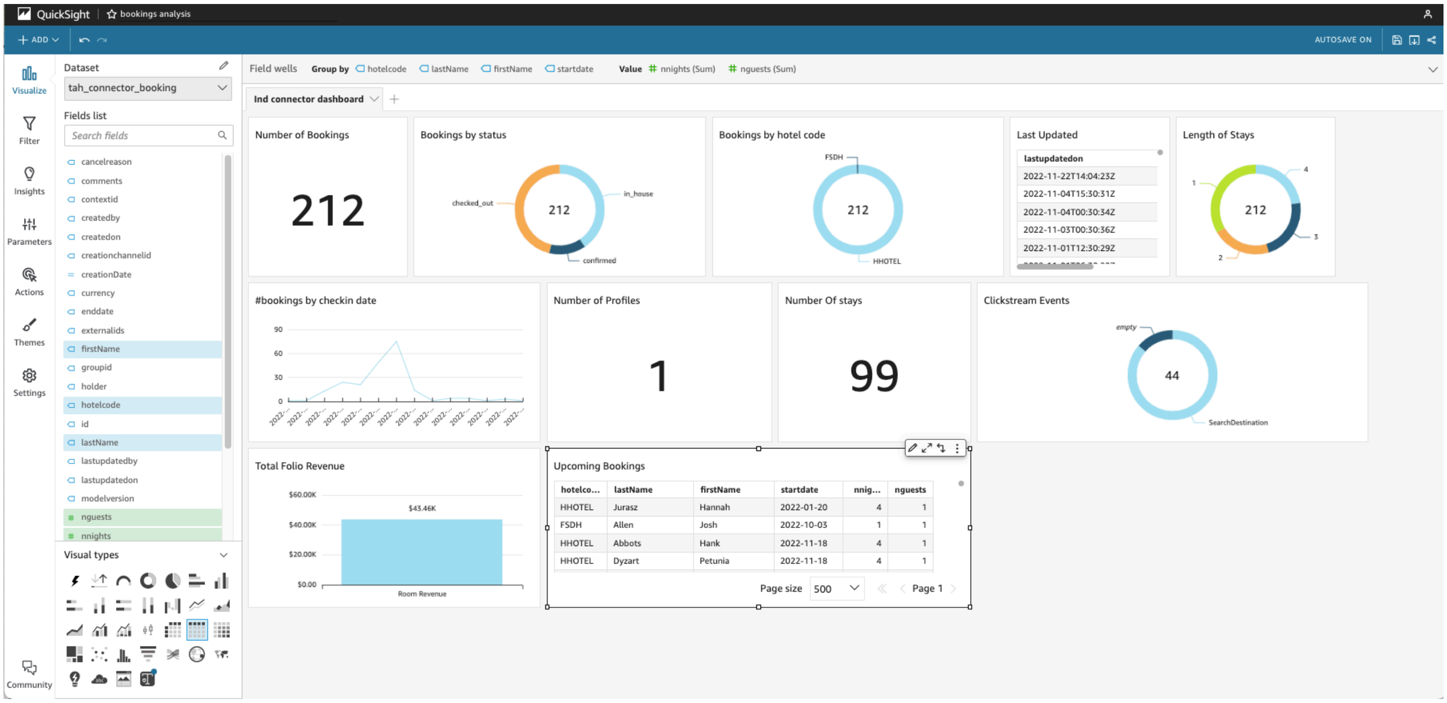
```
SELECT *, holder.firstName, holder.lastName, segments[1].price.totalbeforetax as totalbeforetax
```

- In this example, we access the itemized revenue amounts and type from the folio stay and add it to the data set.

```
SELECT *, rev_item.type, rev_item.amount FROM "tah_ind_connector_db_dev"."stays" CROSS JOIN UNN
```

- You can find more details in the Amazon Athena documentation.

Once you preview the dataset. Select publish and visualize to create your first analysis dashboard. You can now drag and drop widgets and create your own dashboard as illustrated below:



Last Updated

lastupdatedon
2022-11-22T14:04:23Z
2022-11-04T15:30:31Z
2022-11-04T00:30:34Z
2022-11-03T00:30:36Z
2022-11-01T12:30:29Z

Upcoming Bookings

hotelco...	lastName	firstName	startdate	nnig...	nguests
HHOTEL	Jurasz	Hannah	2022-01-20	4	1
FSDH	Allen	Josh	2022-10-03	1	1
HHOTEL	Abbots	Hank	2022-11-18	4	1
HHOTEL	Dyzart	Petunia	2022-11-18	4	1

Page size 500 Page 1

> **Note**
>
> This tutorial illustrates how to create basic visualization of this Connector Solution. It is not meant to be used to create a full-featured data lake analytics. Using this method with very large amount of data may incur large cost as the data is not properly optimized for large and frequent queries. For this advanced usage, you must transform the data into well partitioned PARQUET files using an AWS Glue ETL Job before using it for visualization using Athena. You must also consider Amazon Redshift for very frequent queries as recommended in our Data and Analytics best practices.

Update the solution

If you have previously deployed the solution, follow this procedure to update the Content Localization on AWS CloudFormation stack to get the latest version of the solution's framework.

1. Sign in to the AWS CloudFormation console, select your existing Content Localization 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 latest template.
   c. Paste the link in the **Amazon S3 URL** box.
   d. Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**. 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 10 minutes.

Troubleshooting

Contacting AWS support

Please open a ticket from the AWS console or contact your dedicated Account manager.

Contacting HapiCloud Support

Please contact your HapiCloud representative.

Contacting Tealium Support

Please contact your assigned Customer Success Manager or <`support@tealium.com`>

Uninstall the solution

You can uninstall the Travel and Hospitality Application Connectors Catalog on AWS solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the Amazon S3 buckets created by this solution (see the output of the CloudFormation stack to retrieve the name of the S3 buckets to delete). AWS Solutions Implementations do not automatically delete S3 buckets in case you have stored data to retain (however, the DynamoDB Table will be deleted). Note that two stacks are created by the solution (the code pipeline stack and the solution stack). Both need to be deleted for a full cleanup.

Using the AWS Management Console

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

Using AWS Command Line Interface

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

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

Deleting the Amazon S3 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 S3 buckets. Listed in output of the CloudFormation stack
4. Select the S3 bucket and choose **Delete**.

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

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

Developer guide

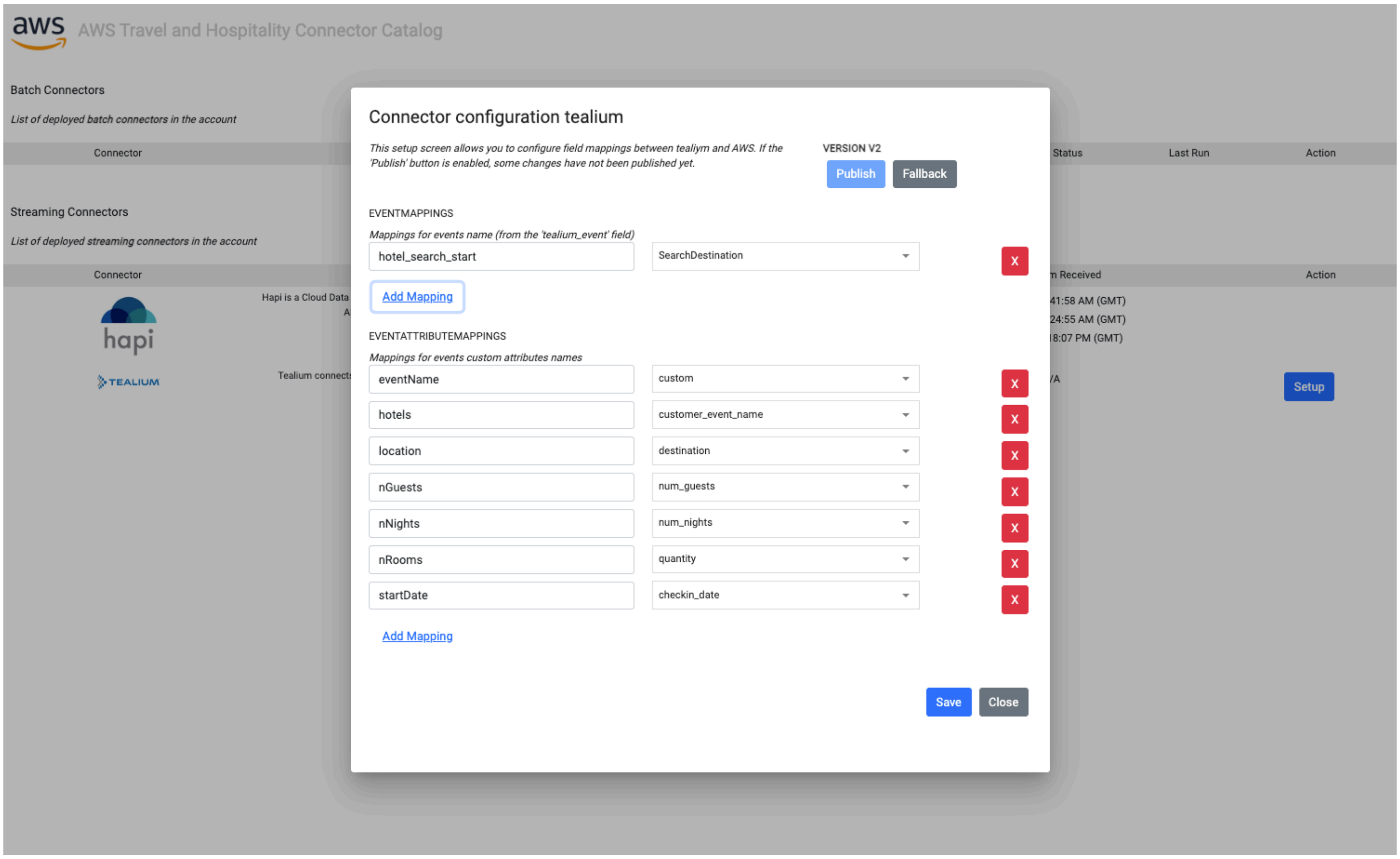
Source code

The Travel and Hospitality Application Connectors Catalog on AWS templates are generated using the AWS Cloud Development Kit (AWS CDK) (AWS CDK).

Set up the Tealium Connector

This section describes the steps required to set up the Tealium connector. It assumes the solution has been successfully deployed as described in the deployment section.

1. Follow the Tealium user guide for setting up the Kinesis firehose connector (authentication required). The Arn of the delivery stream is provided as output of the CloudFormation script and in the email sent following a successful deployment. We recommend to deactivate the Tealium Event Action associated with the connector until Step 3.
2. Log in to the Tealium Administration portal and choose **Setup** to open the mapping modal window. Follow the instruction in the screen to map both the Tealium event name and the Tealium event attribute name to values supported by the AWS data model.
3. Once saved, the mapping must be published to take the live stream into account. The solution provides a fallback feature allowing users to go back to a previous version of the mapping.
4. Once you publish the mapping, go to the Tealium portal and activate the delivery stream. You can monitor the data ingestion using the solution administration portal connector dashboard and analyze further details in the Amazon Data Firehose console.



Connector configuration tealium

This setup screen allows you to configure field mappings between tealiym and AWS. If the 'Publish' button is enabled, some changes have not been published yet.

VERSION V2

EVENTMAPPINGS

Mappings for events name (from the 'tealium_event' field)

hotel_search_start	SearchDestination

EVENTATTRIBUTEMAPPINGS

Mappings for events custom attributes names

eventName	custom
hotels	customer_event_name
location	destination
nGuests	num_guests
nNights	num_nights
nRooms	quantity
startDate	checkin_date

Connector configuration for Tealium mappings

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:

Applicable to all connectors

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each Travel and Hospitality Application Connectors Catalog on AWS solution deployment
- **Timestamp** - Data-collection timestamp
- **Connector** - Name of the connector being utilized
- **Use case name** - Name of the use case executed for this connector
- **Use case run metric** - Metric to be used to aggregate the number of runs for a use case
- **Use case run duration** - Duration of the use case run
- **Status** - Status of the use case run (success or failure)

Hapi

- **Number of objects ingested** - number of objects ingested by the connector
- **Object Type** - Type of object ingested (stay, reservation or profile)
- **Hotel code** - Unique identifier of the hotel in the Hapi system
- **Chain Code** - Unique identifier of the hotel chain in the Hapi system
- **Brand code** - Unique identifier of the brand in the Hapi system

Tealium

- **Number of records** - Number of records ingested from the stream

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

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

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

   to:

   ```
   AnonymousData:
       SendAnonymousData:
         Data: No
   ```
4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, 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 section of this guide.

Contributors

- Geoffroy Rollat
- Owen Brady
- Akash Garg

Revisions

Date	Change
August 2023	Initial release
October 2023	Release version 1.0.1: Updated package versions to resolve security vulnerabilities.
December 2023	Release version 1.1.0: No new features or enhancements. Updated shared schema due to update for the complementary AWS Solution - Unified Profiles for Travelers and Guests on AWS. For additional information, refer to the Unified Profiles for Travelers and Guests on AWS Revisions table.

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.