

# Build Your Embedded Application

## Technical Onboarding Guide

Doyoung Jung, Partner Solutions Architect (dojung@amazon.com)

July 2023 - Last updated: July 2023

**Powered by QuickSight**



## Contents

<b>Section 0 - Overview .....</b>	<b>4</b>
<b>0.1 Document Structure .....</b>	<b>4</b>
0.1.1 Sections.....	4
0.1.2 Revision History .....	4
<b>0.2 Program Overview.....</b>	<b>5</b>
0.2.1 Amazon Partner Network (APN).....	5
0.2.1 Powered by QuickSight Program.....	5
0.2.2 Case Studies.....	6
<b>Section 1 - Data Strategy .....</b>	<b>7</b>
<b>1.1 SaaS Data Storage .....</b>	<b>7</b>
1.1.1 Selecting a Data Store.....	7
1.1.2 Multi-Tenancy Strategy .....	7
<b>1.2 Data Management in QuickSight.....</b>	<b>8</b>
1.2.2 SPICE vs Direct Query .....	8
1.2.3 Data Refresh Requirements .....	9
1.2.4 Row-Level Security .....	10
1.2.5 Column-Level Security.....	11
<b>Section 2 - Creating Dashboards.....</b>	<b>13</b>
<b>2.1 Dashboard Authoring .....</b>	<b>13</b>
2.1.1 Getting Started .....	13
2.1.2 Dashboard General Tips and Tricks .....	13
<b>2.2 Industry Examples .....</b>	<b>15</b>
2.2.1 QuickSight Gallery .....	15
2.2.2 Demo Central.....	15
<b>Section 3 - QuickSight Embedding .....</b>	<b>16</b>
<b>3.1 - Embedding Overview.....</b>	<b>16</b>
3.1.1 “How” to embed.....	16
3.1.2 “What” to embed .....	18
3.1.3 Embedding Demo .....	18
<b>3.2 - 1-Click Embedding .....</b>	<b>18</b>
3.2.1 Enterprise Embedding .....	19
3.2.2 Public Embedding .....	20
<b>3.3 - API-Based Embedding.....</b>	<b>20</b>
3.3.1 Anonymous Embedding .....	20
3.3.2 User-Based Embedding .....	23
<b>3.4 QuickSight Embedding SDK.....</b>	<b>27</b>
3.4.1 Embedding SDK Overview .....	27
3.4.2 Embedding SDK Capabilities .....	28
3.4.2 Embedding SDK Installation .....	29
3.4.3 Resources.....	29
<b>Section 4 – Pricing &amp; Cost Tracking.....</b>	<b>30</b>

- 4.1 QuickSight Pricing.....30**
  - 4.1.1 Primary Cost Factors..... 30
  - 4.1.2 Additional Cost Factors..... 31
- 4.2 Cost & Usage Tracking .....31**
  - 4.2.1 Tracking through the AWS Cost and Usage Reports ..... 31
  - 4.2.2 Resources..... 31
- Section 5 - Getting Help .....32**
  - 5.1 Public Resources.....32**
    - 5.1.1 QuickSight Community ..... 32
    - 5.1.2 QuickSight Documentation..... 32
  - 5.2 AWS Resources .....32**
    - 5.2.1 AWS Support..... 32
    - 5.2.2 Powered by QuickSight Team..... 33

## Section 0 - Overview

Welcome to your Embedded Analytics journey! The Powered by QuickSight Team is excited to assist you and accelerate your development along the way. The purpose of this documentation is to provide you with all the resources you will need to successfully build out an embedded application using Amazon QuickSight.

### 0.1 Document Structure

This documentation aims to address questions that are most commonly asked by our partners when they embed QuickSight. Use this guide as a reference you can scan over as you start your technical validation and build process. It is intended to help you accelerate your understanding of core concepts required to embed QuickSight. If you have suggestions, feedback, or content update requests, please reach out to the Powered by QuickSight Team at [pbq-support@amazon.com](mailto:pbq-support@amazon.com).

#### 0.1.1 Sections

*Section 1 – Data Strategy* will help you determine best practices for 1/ Storing data for your multi-tenant application and 2/ Connecting to and managing that data in QuickSight.

*Section 2 – Creating Dashboards* outlines resources you can use to build interactive dashboards for your end customers.

*Section 3 – Embedding* will help you navigate through the different embedding options and determine the option that best fits your use case.

*Section 4 – Pricing & Cost Tracking* will go over the QuickSight pricing model and help you track QuickSight usage per tenant from a pricing perspective.

*Section 5 – Getting Help* will point you to resources you can rely on when you face issues and require assistance.

#### 0.1.2 Revision History

Since QuickSight is a rapidly growing service, new features additions and documentation changes occur quite frequently. Please refer to this section to track down revisions to this technical onboarding guide.

Date	Change
July 2023	Initial Release

## 0.2 Program Overview

Powered by QuickSight (PbQ) is designed to 1/integrate embedded analytics via QuickSight (QS) to organizations that provide SaaS to their end customers and 2/assist in their go-to-market (GTM) efforts with embedded analytics to their end customers. In this section, learn how AWS can help accelerate your technical evaluation and build steps of incorporating QuickSight into your SaaS application.

### 0.2.1 Amazon Partner Network (APN)

**To fully take advantage of PBQ program benefits, please officially join the AWS Partner Network as a Software Partner.**

Learn more about APN [here](#).

Register to become an AWS Partner [here](#).

### 0.2.1 Powered by QuickSight Program

Powered by QuickSight (PbQ) is a program designed to help AWS Partners build, market, and sell SaaS products that are embedded with Amazon QuickSight. Regardless of where you stand on your embedded analytics journey, the program team provides the support that you will need.

#### **In the Build phase:**

- Engage with Powered by QuickSight technical experts
- Receive a curated technical enablement plan and migration tips
- Introductions to validated Amazon QuickSight embedding specialist consulting partners
- Access to rapid Proofs of Concepts and Workshops

#### **In the Market phase:**

- AWS promotion through LinkedIn, APN blog, AWS Partner Events Hub
- Press release, support for partner led webinar with AWS speaker
- Co-Marketing
  - Customer references, partner Spotlight Webinars, lead gen programs, AWS Marketplace Listing and access to private events & speaking opportunities
- Lead generation programs

#### **In the Sell phase:**

- Opportunity to present at AWS events
- Co-sell support:
  - Assets on how to position BI to your customers
  - Support at your events
- Include AWS experts in your customer engagements
- Amplify your Marketplace, SaaS Factory and ISV Accelerate efforts

To learn more about Powered by QuickSight, watch this 90 second video [here](#).  
If you haven't already, take a minute to register [here](#).

### 0.2.2 Case Studies

If you are curious to hear about successful partnerships between AWS and our partners through the Powered by QuickSight Program, refer to the below resources.

- [Trakstar unlocks new analytical opportunities for its HR customers with Amazon QuickSight](#)

# Section 1 - Data Strategy

This section will help you determine best practices for 1/ Storing data for your multi-tenant application and 2/ Connecting to and managing that data in QuickSight.

## 1.1 SaaS Data Storage

As an ISV, multitenant storage represents one of the more challenging aspects of building and delivering Software as a Service (SaaS) solutions. There are a variety of strategies that can be used to partition tenant data, each with a unique set of nuances that shape your approach to multitenancy.

### 1.1.1 Selecting a Data Store

When choosing a data storage solution, ensure that the data source can be connected to Amazon QuickSight. If QuickSight does not support native connection to your data storage solution, you may need to perform ETL workloads to first load data into a supported data source.

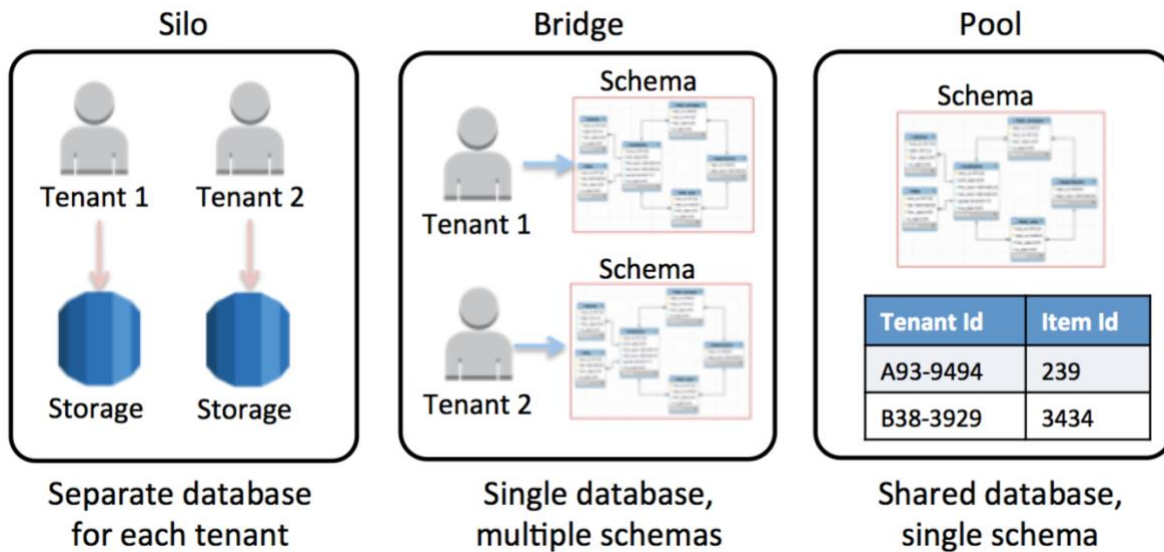
Amazon QuickSight supports a variety of data sources that you can use to provide data for analyses – including native AWS data sources (ex: Amazon S3, Amazon Athena, Amazon RDS, Amazon Redshift, etc.), open-source data sources (ex: MySQL, PostgreSQL, Presto, etc.), and SaaS data sources (ex: Salesforce, Databricks, Snowflake, etc.).

Check the complete list of supported data sources [here](#).

### 1.1.2 Multi-Tenancy Strategy

If you have not already, you will also need to determine the multi-tenancy strategy for your selected data store.

To get started, you need a well-defined conceptual model to help you understand the various implementation strategies. The following figure shows the three basic models—silo, bridge, and pool—that are commonly used when partitioning tenant data in a SaaS environment.



- For more information, read the *AWS Whitepaper for SaaS Storage Strategies* [here](#)
- If using S3, read this blog on *Partitioning and Isolating Multi-Tenant SaaS Data* with Amazon S3 [here](#)

## 1.2 Data Management in QuickSight

Once you have figured out your data storage strategy, the next step is to connect and manage the data in QuickSight.

### 1.2.2 SPICE vs Direct Query

QuickSight Datasets can work in either Direct query or SPICE mode. Keep in mind that query mode can be switched back and forth without impacting the meta data defined in the dataset.

#### Direct Query

In Direct query mode, queries are executed against backend data source each time a dashboard/analysis (that uses the dataset) is loaded.

#### SPICE

In SPICE mode, QuickSight extracts a point in time snap shot of the data - pulling from all tables defined in the dataset per joins specified therein. This extracted data is stored in QuickSight SPICE layer and is used to drive all requests against that dataset. Data in SPICE layer can be refreshed on a schedule (which can be set from UI) or triggered as last step in your data refresh pipeline via create-ingestion API call.

Using SPICE mode gets you two benefits.



1. Scale - QuickSight handles all the load without burdening the underlying data source. So, you don't have to worry if it is 10 users or 10K users hitting a dataset via one or more dashboards.
  2. Speed - SPICE layer is built for QuickSight with performance in mind and provides super-fast dashboard loads.
- SPICE pricing is outlined on Section 4.1.2 – Additional Cost Factors.
  - Learn more about SPICE [here](#).

### 1.2.3 Data Refresh Requirements

You may have data refresh requirements for your customers. The frequency of your data refresh in QuickSight depends on the query mode (SPICE vs Direct Query), as well as the underlying data source you are connecting to.

Learn more about QuickSight Data Refresh [here](#).

#### Direct Query

In direct query mode, your data refreshes whenever a user reloads the dashboard page. When choosing this option, make sure that your data source can handle concurrent queries made without experiencing performance issues. You may also want to consider query costs (for services like Amazon Athena), if applicable.

#### SPICE Refresh

When using SPICE, you can refresh a SPICE dataset manually, or on a schedule. When on a schedule, your data in SPICE can be refreshed Hourly, Daily, Weekly, or Monthly. When the refresh happens, data in SPICE will be updated with new data by querying the underlying data source.

#### SPICE Incremental Refresh

For SQL-based data sources, such as Amazon Redshift, Amazon Athena, PostgreSQL, or Snowflake, you can refresh your data incrementally within a look-back window of time.

An incremental refresh queries only data defined by the dataset within a specified look-back window. It transfers all insertions, deletions, and modifications to the dataset, within that window's timeframe, from its source to the dataset. The data currently in SPICE that's within that window is deleted and replaced with the updates.

With incremental refresh, your data in SPICE can be refreshed every 15 minutes, every 30 minutes, Hourly, Daily, Weekly, or Monthly.

### 1.2.4 Row-Level Security

In many cases, your application will have multiple personas that need access to different levels of data. For example, you probably want to make sure that a user from Company A does not see the same data as a user from Company B. Or, one step further, you may have a case where regional managers from Company A should be able to see regional data, while global managers should be able to see all data.

In these cases, you will want to restrict the data access for your end users using Row-Level Security rules. There are two types of Row-Level Security rules.

#### User-Based Row-Level Security

User-Based RLS requires you to specify a rule dataset that specifies the rules you want to enforce.

User-based rules  ON ^

Choose this to restrict data for users or provisioned groups. Select a permissions dataset with rules for each user or group with access. Only those identified in the rules will be able to see the data. [Learn more](#)

**Selected dataset rules**

user-based-rls

user-based-rls ×

<input type="radio"/>		<input type="checkbox"/>	File	SPICE
<input type="radio"/>		<input type="checkbox"/>	File	SPICE
<input type="radio"/>		<input type="checkbox"/>	File	SPICE
<input checked="" type="radio"/>	user-based-rls	<input type="checkbox"/>	File	SPICE

Permissions Policy:

Grant access to dataset

Remove datasetApply dataset

Below is an example of what the user-based-rls file may look like. While this is a csv file in our case, you can also manage this list dynamically as a database table if needed.

Dataset	GroupARN	customer
	String	String
	arn:aws:quicksight:us-east-1: :group/default/pbq-admin	
	arn:aws:quicksight:us-east-1: :group/AnyFinance/AnyFinance	AnyFinance
	arn:aws:quicksight:us-east-1: :group/AnyGrocery/AnyGrocery	AnyGrocery

Learn more about User-Based RLS [here](#).

### Tag-Based Row-Level Security

Tag-Based RLS requires you to specify the column, tag key. Below is an example of what a Tag-Based RLS setup may look like. In the below example, we are using the customer column as the “company\_name” tag – which is the tag key that will be specified in the API called from your application.

Tag-based rules  ON

Choose this when restricting access to data in embedded dashboards for viewers who are not provisioned in QuickSight. Pass tag values to the QuickSight embedding API to define who sees the data. [Learn more](#)

**Manage tags** Search by column or tag

Column <sup>ⓘ</sup>	Tag	Delimiter <sup>ⓘ</sup>	Match all <sup>ⓘ</sup>	
<input type="text" value="Select column"/>	<input type="text" value="Tag key"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>
customer	company_name	,	*	<input type="button" value="edit"/> <input type="button" value="delete"/>

**Manage rules**

company_name	<input type="button" value="edit"/> <input type="button" value="delete"/>
--------------	---

Learn more about Tag-Based RLS [here](#).

### 1.2.5 Column-Level Security

In some cases, you may have a use case where you want to expose certain columns to a select group of users, while restricting access to everyone else. With column-level security, you can select columns and control which users and groups have access to that column.

### Select the columns you want to restrict

Column-level security lets you manage access to specific columns in your dataset.

Search for a column, user or group   All columns

Page size 50

<input type="checkbox"/>	COLUMN NAME	DATA TYPE	USERS AND GROUPS WITH ACCESS
<input checked="" type="checkbox"/>	row id	Int	Everyone
<input type="checkbox"/>	order id	String	Everyone
<input type="checkbox"/>	order date	Date	Everyone
<input type="checkbox"/>	date key	Int	Everyone
<input type="checkbox"/>	contact name	String	Everyone
<input type="checkbox"/>	country	String	Everyone
<input type="checkbox"/>	city	String	Everyone
<input type="checkbox"/>	region	String	Everyone
<input type="checkbox"/>	subregion	String	Everyone
<input type="checkbox"/>	customer	String	Everyone
<input type="checkbox"/>	customer id	Int	Everyone
<input type="checkbox"/>	industry	String	Everyone
<input type="checkbox"/>	segment	String	Everyone
<input type="checkbox"/>	product	String	Everyone
<input type="checkbox"/>	license	String	Everyone

1 of 20 selected

Learn more about Column-Level Security [here](#).

## Section 2 - Creating Dashboards

This section will guide you through the basics of building dashboards that are interactive and visually appealing.

### 2.1 Dashboard Authoring

#### 2.1.1 Getting Started

To get started with QuickSight dashboard authoring, go through the QuickSight Author Workshop [here](#). This workshop will walk you through how to get started with learning the basics of connecting to data, building visuals, creating calculations, designing interactivity, leveraging ML, and sharing dashboards.

#### 2.1.2 Dashboard General Tips and Tricks

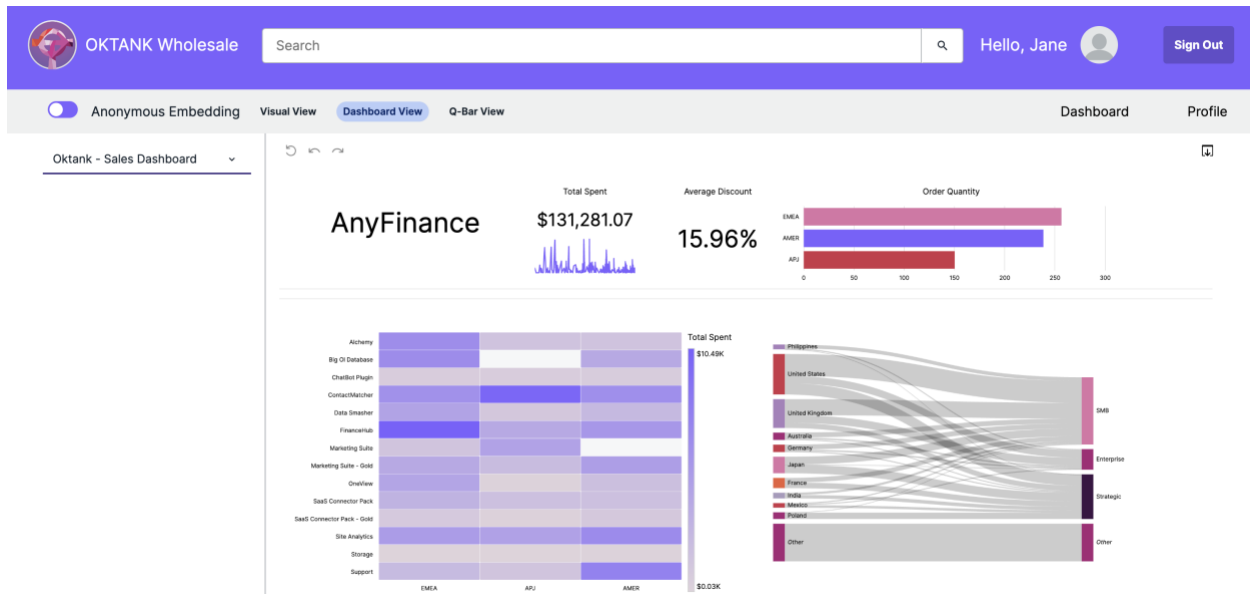
Once you become familiar with dashboard authoring, start using advanced features like themes, layouts, actions, conditional formatting, filters, parameters, and much more.

Learn more about Tips and tricks for high-performant dashboards in Amazon QuickSight [here](#).

#### Themes

In Amazon QuickSight, a theme is a collection of settings that you can apply to multiple analyses and dashboards. You can create your own theme to match the color scheme of your host application.

In the example below, you can see that the theme of the Sales Dashboard matches the theme of the host application. This creates a seamless user experience for the end-user.



Learn more about Themes [here](#).

## Layouts

There are three dashboard layout designs you can choose from: Tiled, Free-form, and Classic. For simple dashboards, start off by using the tiled layout. For dashboards that require more customization, use free-form layout for maximum creativity.

1. Tiled layout
  - a. Visuals in a Tiled layout snap to a grid with standard spacing and alignment.
  - b. You can make visuals any size and place them wherever you want within a dashboard, but visuals can't overlap.
2. Free-form layout
  - a. Visuals in a Free-form layout can be placed anywhere in your dashboard using precise coordinates.
  - b. You can drag a visual to the exact place you want, or you can enter the coordinates of the visual's location.
3. Classic layout
  - a. Visuals in a Classic layout snap to a grid with standard spacing and alignment.
  - b. Dashboards hide data or change formatting to fit smaller screen sizes.
    - i. For example, if you change a visual to make it considerably smaller, the on-visual menu and editors are hidden so that the chart elements have more room to display.
    - ii. Bar chart visuals can also display fewer data points.

Learn more about Layouts [here](#).

## 2.2 Industry Examples

One of the best ways to learn is to look at example dashboards built by internal experts, as well as customers and partners that are already using QuickSight for their BI workload.

### 2.2.1 QuickSight Gallery

[QuickSight Gallery](#) is a publicly accessible resource where you can explore sample industry and domain specific analytics built by Amazon QuickSight customers. Check out example dashboards from various industries like retail, healthcare, manufacturing, and more.

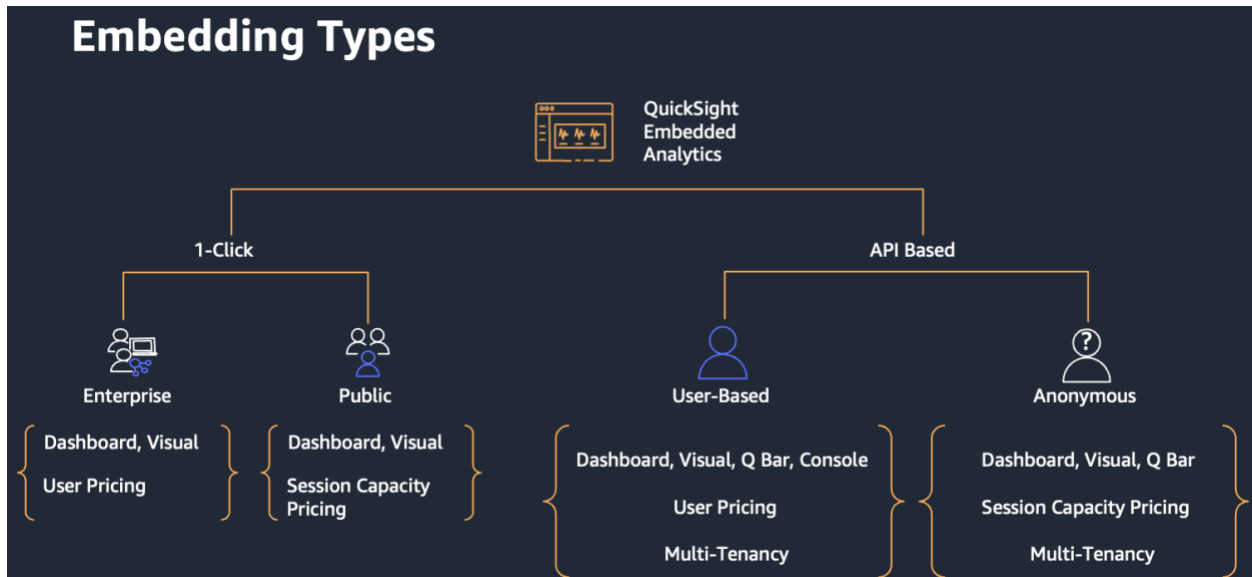
### 2.2.2 Demo Central

[Demo Central](#) is a sandbox environment for you to test out QuickSight features. You can explore and interact with pre-built dashboards created by our specialists, as well as test out features like calculations, RLS (Row-Level-Security), drill-downs, and more.

## Section 3 - QuickSight Embedding

This section will help you navigate through the different embedding options and determine the option that best fits your use case. Keep in mind that all embedding features require QuickSight Enterprise Edition.

### 3.1 - Embedding Overview



Embedding Types	Registered (User-Based)	Not-Registered (Anonymous)
1-Click	<b>Enterprise</b> <ul style="list-style-type: none"> <li>Dashboard, Visual</li> <li>User Pricing</li> </ul>	<b>Public</b> <ul style="list-style-type: none"> <li>Dashboard, Visual</li> <li>Session Capacity Pricing</li> </ul>
API-Based	<b>User-Based</b> <ul style="list-style-type: none"> <li>Dashboard, Visual, Console, Q Bar</li> <li>Multi-tenancy (Namespace + User RLS)</li> <li>User Pricing</li> </ul>	<b>Anonymous</b> <ul style="list-style-type: none"> <li>Dashboard, Visual, Q Bar</li> <li>Multi-tenancy (Tag-Based RLS)</li> <li>Session Capacity Pricing</li> </ul>

#### 3.1.1 "How" to embed

QuickSight supports two embedding methods:

- 1-Click Embedding



- a. This is the quick and easy embedding method that doesn't require any programming. You select the dashboard or visual you want to embed, and click the "embed" button that automatically generates an iframe element for you
  - i. In general, 1-Click embedding is great for simple use-cases where there are no complex requirements such as multi-tenancy
- b. Types
  - i. 1-Click Embedding can be further broken down into Enterprise vs Public Embedding, depending on how end-users are charged and authenticated
    - 1. Enterprise Embedding**
      - a. In this case, you are an enterprise with internal QuickSight users. You have an internal application or portal where you want to embed dashboards for internal users
      - b. When users try to access the embedded dashboard or visual, they will either be asked to sign into QuickSight or be automatically authenticated through SSO (if enabled on the application)
    - 2. Public Embedding**
      - a. In this case, you have a public facing website and you want everyone with access to your website (potentially everyone on the internet) to be able to view and interact with the embedded dashboard or visual
      - b. When users try to access the embedded dashboard or visual, they will consume one "session". No further authentication is required
      - c. Public Embedding requires that you have session capacity pricing enabled, which is a pricing option where you purchase "sessions" upfront for your users to consume
- 2. API-Based Embedding
  - a. This is the embedding method that calls the `GenerateEmbedUrlForRegisteredUser` or `GenerateEmbedUrlForAnonymousUser` APIs to generate the embed URL
    - i. In general, API-Based embedding is the more commonly used embedding method since most ISVs have complex use cases
  - b. Types
    - i. API-Based Embedding can be further broken down into User-Based vs Anonymous Embedding, depending on the user pricing option
      - 1. User-Based Embedding**
        - a. In this case, you create a 1-to-1 mapping of your application users and QuickSight. Essentially, you create every application user as a new user in your QuickSight account
      - 2. Anonymous Embedding**
        - a. In this case, users don't need to be authenticated into QuickSight to view the embedded resource

- i. Instead, they consume one session when they interact with the embedded resource
- b. Anonymous Embedding requires that you have session capacity pricing enabled, which is a pricing option where you purchase “sessions” upfront for your users to consume

### 3.1.2 “What” to embed

QuickSight supports embedding for these elements:

1. QuickSight console (only available for User-Based Embedding)
2. QuickSight dashboards (available for User-Based, Anonymous, Enterprise, and Public Embedding)
3. QuickSight visuals (available for User-Based, Anonymous, Enterprise, and Public Embedding)
4. QuickSight Q search bar (only available for User-Based Embedding and Anonymous Embedding)

### 3.1.3 Embedding Demo

If you want to see a demo of embedding, refer to the resources below.

- [Oktank Wholesale Application Demo](#)
  - This is a demo application created by the Powered by QuickSight Team
  - <https://github.com/aws-samples/amazon-quicksight-embedding-code-sample>
    - You may log in as two different users from two different tenants
      - [jane@anyfinance.com](mailto:jane@anyfinance.com)
        - PW: Demo1234!
      - [diego@anygrocery.com](mailto:diego@anygrocery.com)
        - PW: Demo1234!
- [QuickSight Embedded Analytics Developer Portal](#)
- [Anysoftware.com Demo](#)
- [QuickSight Embedding Demo](#)

## 3.2 - 1-Click Embedding

1-Click Embedding, as the name suggests, is the simplest form of embedding that allows dashboard and visual embedding with a click of a button. If you are an ISV looking for API-Based embedding options, feel free to skip this section.

Learn more about 1-Click Embedding [here](#).

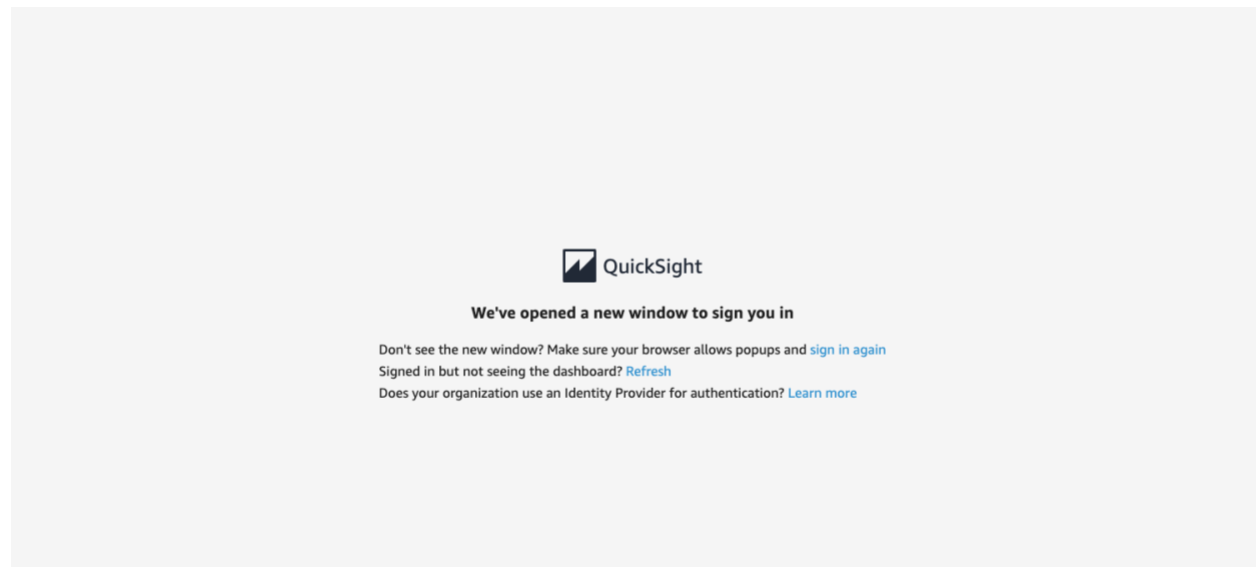
### 3.2.1 Enterprise Embedding

You can embed a visual or dashboard in your internal application for registered users of your Amazon QuickSight account. You do so using the embed code that you get when you share the dashboard or from the Embed visual menu in QuickSight. You don't have to run the QuickSight embedding API to generate the embed code. You can copy the embed code from QuickSight and paste it in your internal application's HTML code.

#### 1-Click Enterprise Embedding Use Case

Using 1-click enterprise embedding is best for when you want to embed a QuickSight dashboard in an internal application that users need to authenticate in to. When you copy the embed code, you get a static embed code that doesn't change.

When users try to access the embedded dashboard or visual, they will either be asked to sign into QuickSight unless they are automatically authenticated through SSO (if enabled on the application).



#### Checklist

Use case	Embedding for an internal application
Authentication	SSO or QuickSight Login
Reader Pricing	User Pricing
Embedding	Dashboards, Visuals

### 3.2.2 Public Embedding

You can embed a visual or dashboard in public sites using the embed code that you get when you share the visual or dashboard in Amazon QuickSight. You can also turn on public sharing by using the QuickSight console and automatically grant access to a shared visual or dashboard to anyone on the internet. After you embed the visual or dashboard in your application, anyone who can access your application can access the embedded visual or dashboard.

<https://catalog.us-east-1.prod.workshops.aws/workshops/f89209a9-398f-4a06-a1de-5b7b0424bbaa/en-US/exercise>

#### 1-Click Public Embedding Use Case

Using 1-click public embedding is best for when you want to embed a QuickSight dashboard into a public website with many viewers. When you copy the embed code, you get a static embed code that doesn't change.

#### Checklist

Use case	Embedding for public site
Authentication	No QuickSight Authentication
Reader Pricing	Capacity Pricing
Embedding	Dashboards, Visuals

## 3.3 - API-Based Embedding

Whereas 1-click embed allows for quick and easy embedding for simple use cases, API-Based embedding gives you more control and flexibility to support complex use cases.

### 3.3.1 Anonymous Embedding

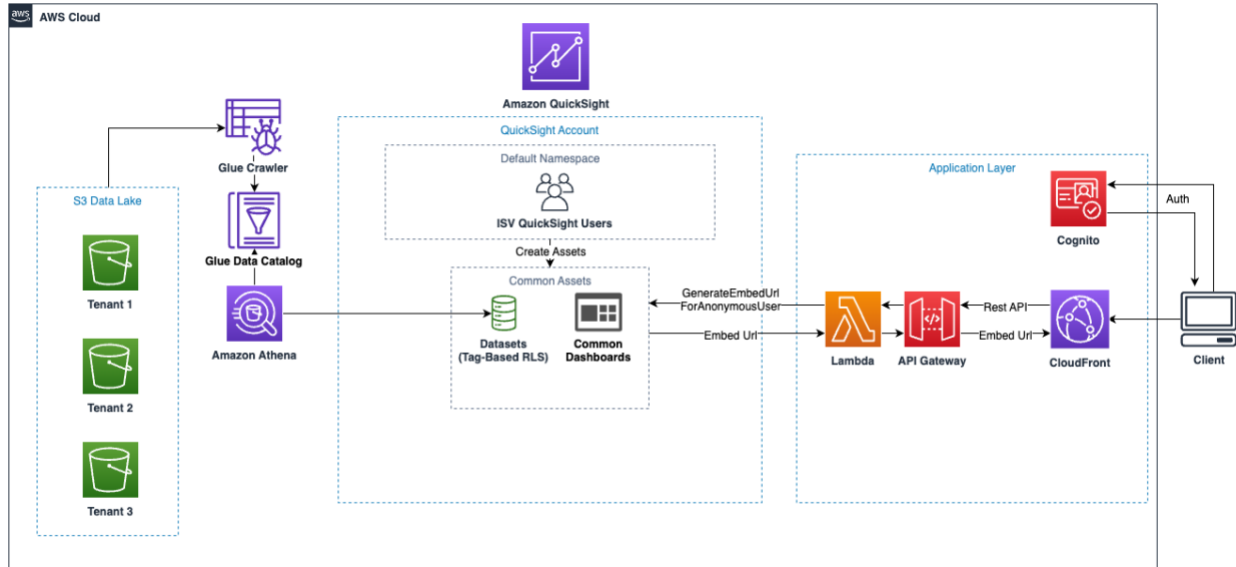
With Anonymous Embedding, users don't need to be authenticated into QuickSight to view the embedded resource. Instead, they consume one session when they interact with the embedded resource. Anonymous Embedding requires that you have session capacity pricing enabled, which is a pricing option where you purchase "sessions" upfront for your users to consume.

To learn more, follow along the Anonymous Embedding Workshop [here](#).

#### Use Case

Anonymous Embedding is recommended for large-scale embedded use cases. Unlike User-Based embedding, this option does not support console embedding.

#### Architecture



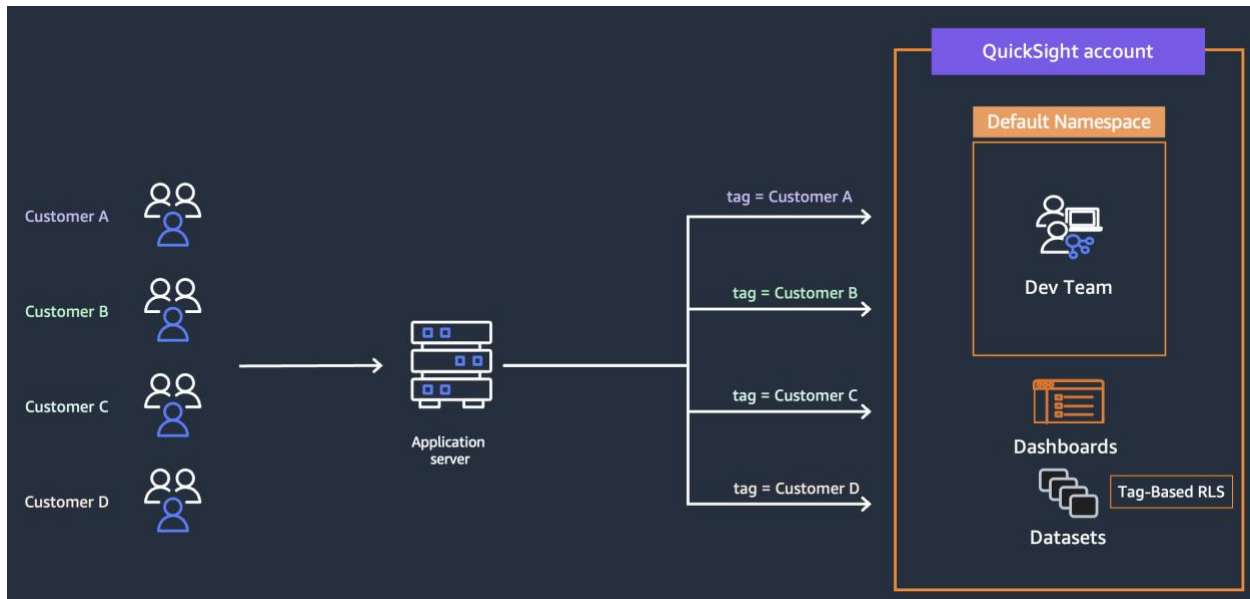
In this sample architecture, our tenant data is stored and partitioned in S3. We catalog this data in the Glue Data Catalog, and use Amazon Athena to connect the S3 data with QuickSight.

In our QuickSight account, we have the ISV’s QuickSight users create dashboards from the Athena dataset. This dataset has Tag-Based RLS enabled for multi-tenancy.

From the application level, a user is first authenticated into the application through an Identity Provider. When the user accesses a page with embedded dashboards, the application extracts the user’s tenant data and passes it into the GenerateEmbedUrlForAnonymousUser API.

QuickSight takes this API request, filters the data with the tag-value that is passed into the API, and returns an embed URL to the application. Once this url is opened on the application, an embedding session is established between QuickSight and the application.

### Multitenancy for Anonymous Embedding



With anonymous embedding, you enforce multi-tenancy by using Tag-Based RLS. You enable tag-based RLS on the QuickSight dataset you are using, and dynamically pass in tenant-specific tags to the API parameter. This allows QuickSight to filter the data that tenants see based on the tag value that is passed.

## API

Anonymous embedding uses the *GenerateEmbedUrlForAnonymousUser* API.

This generates an embed URL that you can use to embed an Amazon QuickSight dashboard or visual in your website, without having to register any reader users. Before you use this action, make sure that you have configured the dashboards and permissions.

Below are the parameters used in the API:

- **AwsAccountId**
  - The ID for the AWS account that contains the dashboard that you're embedding.
- **AuthorizedResourceArns**
  - The Amazon Resource Names (ARNs) for the Amazon QuickSight resources that the user is authorized to access during the lifetime of the session. If you choose Dashboard embedding experience, pass the list of dashboard ARNs in the account that you want the user to be able to view. Currently, you can pass up to 25 dashboard ARNs in each API call.
- **ExperienceConfiguration**
  - The configuration of the experience that you are embedding. You can specify here whether you want to embed a dashboard, visual, or the q search bar.
- **Namespace**

- The Amazon QuickSight namespace that the anonymous user virtually belongs to. Using virtual namespaces for anonymous users makes it easy to track costs.
- SessionTags
  - The session tags used for row-level security. Before you use this parameter, make sure that you have configured the relevant datasets using the `DataSet$RowLevelPermissionTagConfiguration` parameter so that session tags can be used to provide row-level security.

Learn more about the *GenerateEmbedUrlForAnonymousUser* API [here](#)

### Checklist

Use case	Embedding for large-scale applications; No dashboard authoring capabilities for end-user
Authentication	None
Reader Pricing	Session Capacity Pricing
Embedding	Dashboards, Visuals, Q Search Bar

### 3.3.2 User-Based Embedding

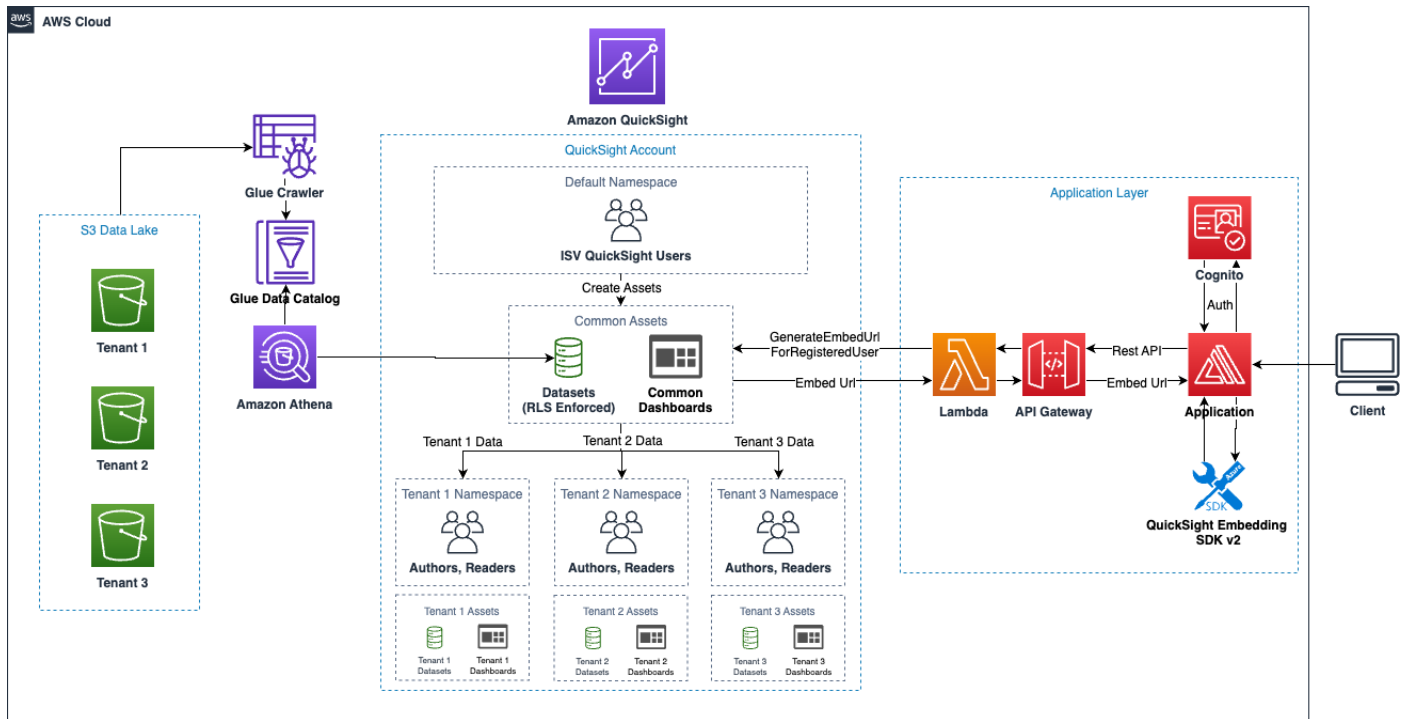
With User-Based Embedding, you create a 1-to-1 mapping of your application users and QuickSight. Essentially, you create every application user as a new user in your QuickSight account

To learn more, follow along the User-Based Embedding Workshop [here](#).

### Use Case

User-Based Embedding is recommended for embedded use cases with high usage per user. It is also the only embedding option that allows console embedding for the embedded authoring experience.

### Architecture

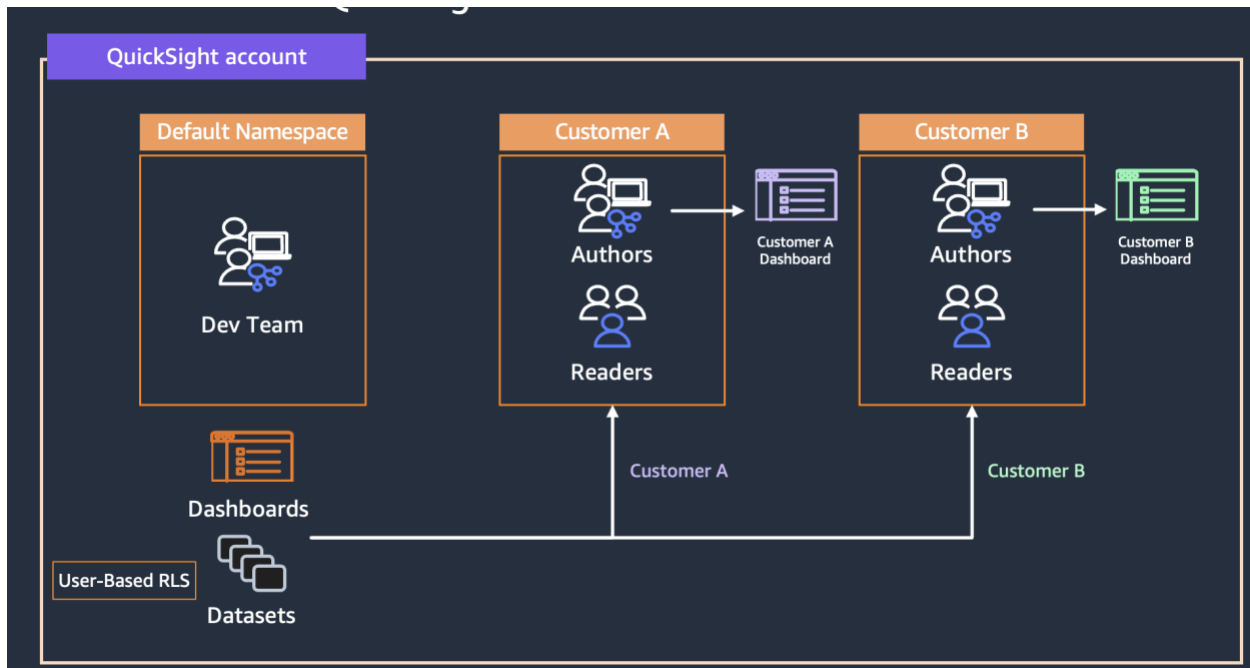


The workflow is as follows:

1. A new user is onboarded to your SaaS
2. Your backend functions ensure that there is an existing QuickSight namespace for your user’s tenant in your QuickSight account.
  - a. If there isn’t an existing namespace for the tenant, your application creates a new namespace using the CreateNamespace API
3. Your backend function registers the new user into the QuickSight using the RegisterUser API
  - a. Make sure that the new user is created under the namespace they belong in
  - b. Determine if the user will be a Reader or Author
  - c. Determine if the user will belong in any Groups
  - d. Determine if the user will be applied any Custom Permission
4. Share relevant dashboards/analyses/datasets are shared with the newly provisioned QuickSight user
  - a. Resources can either be shared programmatically or manually through the QuickSight console

### Multi-tenancy for User-Based Embedding





With user-based embedding, you enforce multi-tenancy with a combination of Namespaces and User-Based Row-Level Security.

A QuickSight namespace is a logical container that you can use to organize clients, subsidiaries, teams, and so on. Namespaces can help you achieve the following goals:

- You can allow the users of your QuickSight subscription to discover shared content and share with other users. At the same time, you can be sure that users in one namespace can't see or interact with users in another namespace.
- You can securely isolate data and also support diverse workloads without adding additional AWS accounts. Access to data is still strictly controlled by AWS security features. Users can see assets (like data and dashboards) only if they have the correct resource permissions. Also, users who have permissions can't inadvertently expose content to people who outside of their namespace.
- You can monitor data flows and usage reports, neatly partitioned by namespace. Categorizing data and reports by namespace can help simplify cost and security analysis.
- After you've registered users into your namespace, there's no additional administrative complexity or overhead.

Learn more about namespace [here](#).

### Custom Permissions

Embedding the QuickSight console gives your users access to the QuickSight console. In many cases, you may want to restrict certain user actions inside the console - such as the ability to add new data sources or running features like anomaly detection.

With custom permissions, you can specify which actions you want to restrict, and apply the rule to your users.

### Edit custom permissions

Name

pbq-tenant

#### Restrict access to

##### Dashboard & analyses

- Adding or running anomaly detection
- Creating or updating themes
- Exporting to CSV
- Exporting to Excel
- Sharing analyses
- Sharing dashboards
- Sharing datasets

##### Datasets

- Creating or updating all datasets
- Creating or updating only SPICE datasets

##### Data sources

- Creating or updating all data sources

##### Folders

- Creating shared folders
- Renaming shared folders

##### Reports

- Creating or updating email reports
- Subscribing to email reports

##### Threshold Alerts

- Creating or updating threshold alerts

Update

Learn more about custom permissions [here](#).

## API

User-Based embedding uses the *GenerateEmbedUrlForRegisteredUser* API.

This generates an embed URL that you can use to embed an Amazon QuickSight dashboard or visual in your website for a specific QuickSight user. Before you use this action, make sure that you have configured the dashboards and permissions, and that the application user is first registered in QuickSight.

Below are the parameters used in the API:

- **AwsAccountId**
  - The ID for the AWS account that contains the dashboard that you're embedding.

- ExperienceConfiguration
  - The configuration of the experience that you are embedding. You can specify here whether you want to embed a dashboard, visual, the q search bar, or the console.
- UserArn
  - The Amazon Resource Name for the registered user
  - Here is an example –
- AllowedDomains
  - The domains that you want to add to the allow list for access to the generated URL that is then embedded. This optional parameter overrides the static domains that are configured in the Manage QuickSight menu in the Amazon QuickSight console. Instead, it allows only the domains that you include in this parameter. You can list up to three domains or subdomains in each API call.
- SessionLifetimeInMinutes
  - How many minutes the session is valid. The session lifetime must be in [15-600] minutes range.

Learn more about the *GenerateEmbedUrlForRegisteredUser* API [here](#)

### Checklist

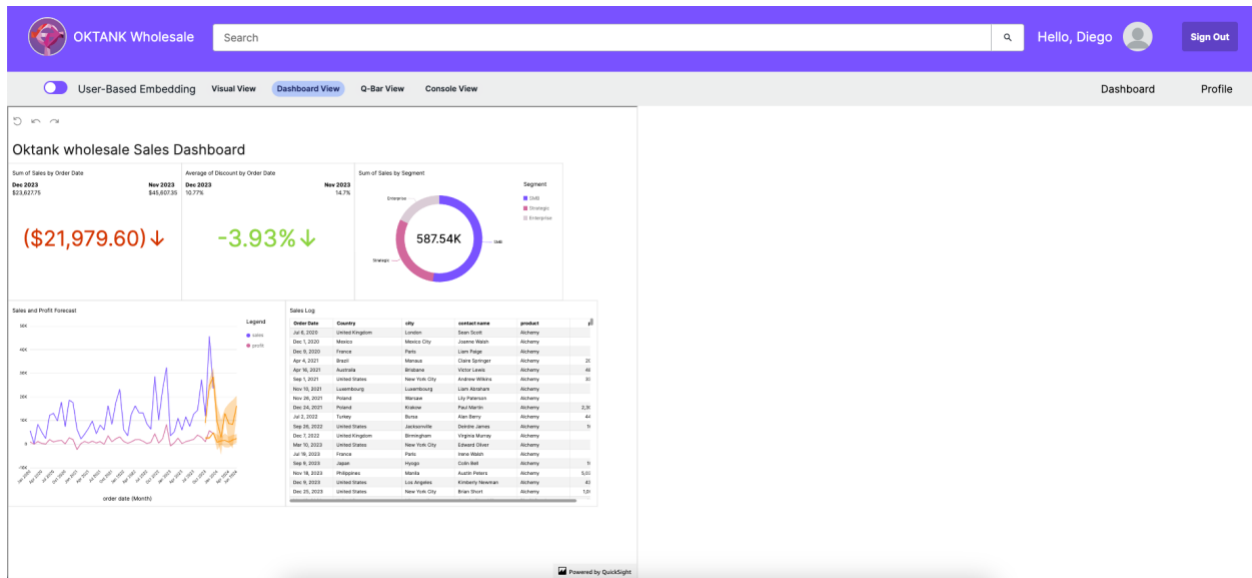
Use case	Embedding for application with high usage per user; Embedding for dashboard authoring
Authentication	1-to-1 user mapping between application and QuickSight
Reader Pricing	User-Based Pricing
Embedding	Dashboards, Visuals, Q Search Bar, Console

## 3.4 QuickSight Embedding SDK

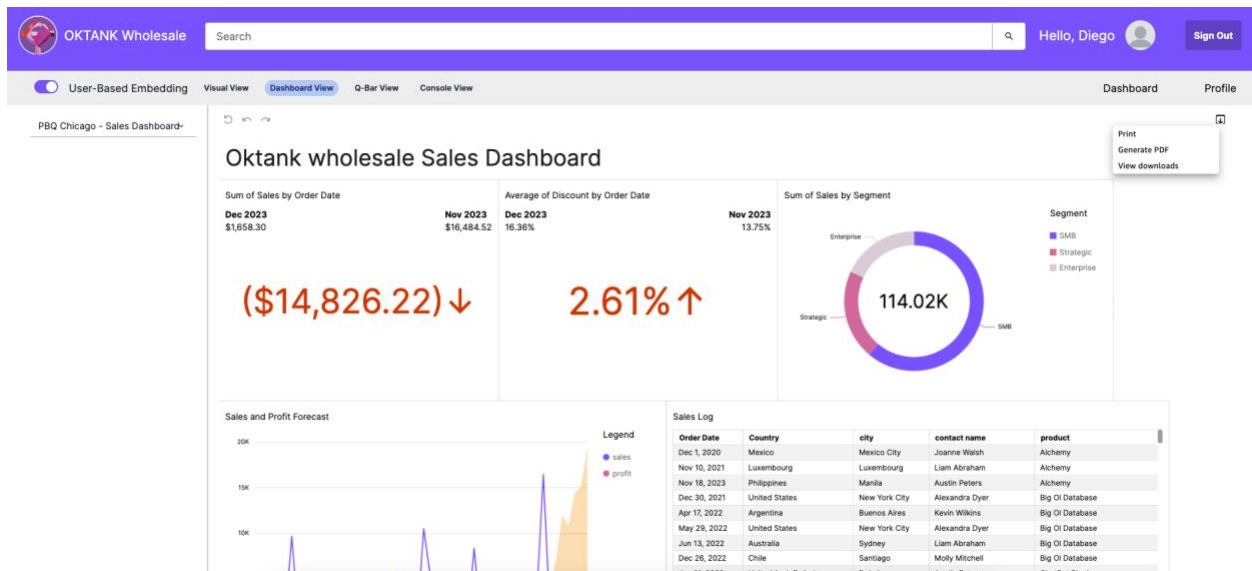
When embedding with the API, you can customize the session by using the QuickSight Frontend Embedding SDK.

### 3.4.1 Embedding SDK Overview

Once you have generated the Embed Url, you can embed the url into your front-end application using an iframe. In the below screenshot, I am embedding the iframe into my application using vanilla HTML code. While we can see the dashboard, you may notice that the iframe is not very visually appealing.



With the Embedding SDK, you can customize the embedded iframe and add interactivity to it as you can see in the screenshot below. Much more appealing!



### 3.4.2 Embedding SDK Capabilities

The embedding SDK can be used for a variety of use cases, including but not limited to:

- Showing/Hiding tool bar
- Customizing width and height of iframe
- Render a loading animation while the dashboard loads
- Monitor interactions in embedded dashboards
- Initiate dashboard print from the application

- State persistence
- State persistence is currently supported only for the user-based (not anonymous) dashboard and console embedding experience.
- Passing in parameters from the application into dashboards

See the official documentation for QuickSight Embedding SDK [here](#).

### 3.4.2 Embedding SDK Installation

**Option 1:** Use the Amazon QuickSight Embedding SDK in the browser:

```
...  
<script src="https://unpkg.com/amazon-quicksight-embedding-  
sdk@2.2.1/dist/quicksight-embedding-js-sdk.min.js"></script>  
<script type="text/javascript">  
  const onLoad = async () => {  
    const embeddingContext = await QuickSightEmbedding.createEmbeddingContext();  
    //...  
  };  
</script>  
...  
<body onload="onLoad()">  
...
```

**Option 2:** Install the Amazon QuickSight Embedding SDK in NodeJs:

```
npm install amazon-quicksight-embedding-sdk
```

### 3.4.3 Resources

- [Enhance your analytics embedding experience with the new Amazon QuickSight JavaScript SDK](#)
- [Level up your React app with Amazon QuickSight: How to embed your dashboard for anonymous access](#)

## Section 4 – Pricing & Cost Tracking

### 4.1 QuickSight Pricing

#### 4.1.1 Primary Cost Factors

Primary Cost Factors for QuickSight are Authors, Administrators, and Readers

##### Author/Administrators pricing

- **Month-to-Month Commitment**
  - \$24 per month
    - For Q-enabled accounts. \$34 per month
- **Annual Commitment**
  - \$18 per month
    - For Q-enabled accounts, \$28 per month

##### Reader pricing

QuickSight offers two reader pricing models: User pricing and Capacity pricing.

- **User Pricing**
  - Individual user-based pricing which is ideal for Business Intelligence (BI) applications or embedded use cases with high usage per user
  - \$0.30 per session for 30 mins, up to a maximum of \$5 per reader per month
    - A “session” is initiated by Logging in to the QuickSight console or via embedded dashboards
      - Not charged for additional session if inactive (i.e., sitting in the background)
    - For Q-enabled accounts, the \$5 per month max becomes a \$10 per month max (which is reached on the 34th monthly session)
- **Capacity Pricing**
  - Capacity pricing allows QuickSight customers to purchase reader sessions or Q questions in bulk, without having to provision individual readers in QuickSight. Capacity pricing is ideal for embedded applications or large-scale BI deployments.
  - This is a tiered pricing model, starting with a monthly plan of \$250 per month which includes 500 sessions, and scaling to an annual option of 1,600,000 annual sessions for a commitment of \$258,000 (\$0.16125/session)
  - This is billed monthly based on usage, with a potential true up at the end of the commitment if there is a shortfall in sessions consumed.

#### 4.1.2 Additional Cost Factors

- **QuickSight Q:**
  - Natural language query service in QuickSight
  - Base fee of \$250 per region monthly, so \$3,000 annually
    - Author costs increase \$10 per month (as indicated above)
    - Readers are billed the same (\$0.30/session) but instead of being capped at \$5 per month, the cap is \$10 per month
- **SPICE**
  - In-memory calculation engine/cache
  - Each Author/Administrator license receives 10GB of SPICE capacity which is pooled at the account level
    - 50 Authors would mean 500GB free SPICE capacity at the account level
  - Additional SPICE capacity is \$0.38/GB per month
- **Paginated Reporting**
  - Feature for static, paginated, report distribution
  - This is an optional add on and starts at \$500 per month
- **Alerts/Anomaly Detection**
  - Tiered pricing based on Metrics Evaluated
    - First tier (1 to 1,000,000) is \$.50 per 1,000 metrics evaluated
    - Highest tier is over 100,000,000 and is \$.05 per 1,000 metrics evaluated
  - This is generally a very minimal cost

Learn more about QuickSight Pricing [here](#).

## 4.2 Cost & Usage Tracking

Once you successfully embed QuickSight into your application, you may want to track the usage patterns of your customers and see a breakdown of cost per tenant or per user.

### 4.2.1 Tracking through the AWS Cost and Usage Reports

Administrators can access user level cost and usage data for all user types in AWS Billing Cost and Usage Reports in a standardized format. This allows you to create standardized cost breakdowns of your deployments, and enable cost charge back to customers.

As this is a new feature as of June 2023, more content is being created in this space.

### 4.2.2 Resources

Read the Cost and Usage Tracking blog here. (In-Progress)

Watch the Amazon QuickSight Learning Series video for Advanced QuickSight Cost and Usage analysis [here](#).

## Section 5 - Getting Help

Embedding QuickSight into your existing architecture may require many different pieces to work together. Refer to the below resources if you need help.

### 5.1 Public Resources

#### 5.1.1 QuickSight Community

QuickSight Community is the place for QuickSight users to come together and talk about QuickSight. It is the one-stop shop for answers to questions, tips & tricks, show & tell, and all-things QuickSight. QuickSight Community should be your first go-to when you need answers to quick questions you may run into.

- Check out QuickSight Community [here](#).
- For embedding related questions on QuickSight Community, check [here](#).

#### 5.1.2 QuickSight Documentation

Amazon QuickSight has a number of documentations tailored towards different audiences.

- [User Guide](#) describes all Amazon QuickSight concepts and features, and provides instructions on using these features in the Amazon QuickSight web application.
- [Developer Guide](#) provides usage examples of API operations for Amazon QuickSight and procedural walkthroughs of common tasks.
- [API Reference](#) provides descriptions, syntax, example requests, and sample responses for all of the operations available in Amazon QuickSight, plus details on data types and errors.
- [Amazon QuickSight in the AWS CLI Reference](#) describes all the command line interface (CLI) commands for Amazon QuickSight in detail. Provides all syntax. Also provides examples for the most common commands.

### 5.2 AWS Resources

#### 5.2.1 AWS Support

Depending on your AWS Support Plan, you may have access to AWS Support that provides a mix of tools and technology, people, and programs designed to proactively help customers optimize performance, lower costs, and innovate faster.

AWS Technical Support tiers cover development and production issues for AWS products and services, along with other key stack components:

- "How to" questions about AWS services and features



- Best practices to help you successfully integrate, deploy, and manage applications in [the cloud](#)
- Troubleshooting API and AWS SDK issues
- Troubleshooting operational or systemic problems with AWS resources
- Issues with our Management Console or other AWS tools
- Problems detected by EC2 health checks
- A number of third-party applications such as OS, web servers, email, VPN, databases, and storage configuration

AWS Support does not include:

- Code development
- Debugging custom software
- Performing system administration tasks
- Database query tuning
- Cross-Account Support (for more information, see [AWS Support FAQs](#))

Learn more about AWS Support [here](#).

### 5.2.2 Powered by QuickSight Team

For all other inquiries that cannot be resolved with the resources above, please reach out to the Powered by QuickSight Team at [pbq-support@amazon.com](mailto:pbq-support@amazon.com).