aws

User Guide

# Amazon FinSpace

# Amazon FinSpace: User Guide

# Table of Contents

# What is Amazon FinSpace?

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace is a fully managed data management and analytics service that makes it easy to store, catalog, and prepare financial industry data at scale.

FinSpace removes the heavy lifting of building and maintaining a data management system for financial analytics. FinSpace provides a Managed kdb Insights analytics engine powered by the industry recognized kdb analytics engine. It also features the Dataset browser that you can use to collect data and catalog it by relevant business concepts such as asset class, risk classification, or geographic region; which makes it easy to discover and share across your organization.

To see all the regions FinSpace is available in, visit the AWS Region page.

## Managing data in financial services

Financial services customers store petabytes of data which is collected from both internal and external data sources. The data is generated from their internal applications such as portfolio management systems, actuarial applications, order and risk management systems, and external data such as stock exchanges and financial data providers. The data is typically used for use cases including but not limited to quantitative research, product pricing, customer experience, and investment management. The size of the data is growing and the number of sources that FSIs are receiving data from is also increasing which makes it hard to manage and track. FSI customers want to make this data available in a self-service and secure way to their analysts and data scientists for analysis. The analysts want to discover the data easily, and analyze it at scale.

# Benefits of Amazon FinSpace with Managed kdb Insights

**With Managed kdb, you can:**

- Eliminate operational overhead with Managed kdb Insights for high performance capital markets analytics.
- Manage spend, keep up with market volatility, and ensure high-availability with auto scaling and multi-Availability Zone for kdb applications.
- Launch new analytics infrastructure on demand and accelerate migration of on-premise kdb systems to AWS.

# Benefits of Amazon FinSpace Dataset browser

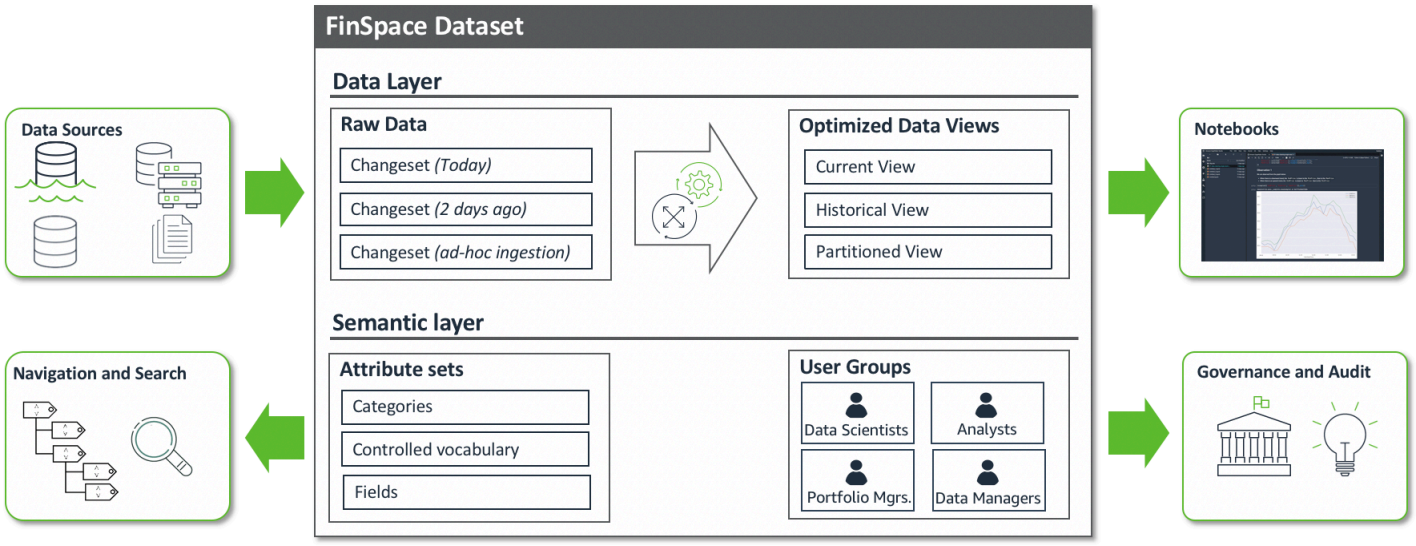
With Amazon FinSpace Dataset browser, you can:

1. **Import data easily** – The SDKs allows you to load data files into FinSpace in bulk, daily, or ad-hoc fashion. Connect your daily historical data feeds from stock exchanges and data providers into FinSpace. For more information, see Loading and analyzing data.
2. **Store and catalog data with business terms** – Create a business data catalog with your business taxonomy to organize data so that your business users can easily discover it. Organize data by asset classes, regions, data types, or industry. For more information, see Configuring a business data catalog.
3. **Track versions of data** – Create bi-temporal views that let you analyze data the way it looked at a particular date and time. Reproduce historical financial models for audit and compliance purposes.
4. **Prepare and analyze data at scale** – Use FinSpace notebook with integrated managed Spark clusters to run analysis on petabytes of data. Scale compute with spark clusters on an as-needed basis. For more information, see Prepare and analyze data.
5. **Share data managed in FinSpace** – Share data view tables with a Lake Formation data lake so that the data can be easily queried with AWS analytics engines like Amazon Redshift, Athena, Amazon QuickSight,Amazon EMR, and SageMaker. For more information, see Data views sharing.
6. **Financial time series analysis** – Run financial time series analysis on high density market data using integrated time series library with over 100 embedded functions including statistical and technical indicators such as Bollinger Bands. For more information, see Time series library.

# Core concepts and terms

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

## Loading data



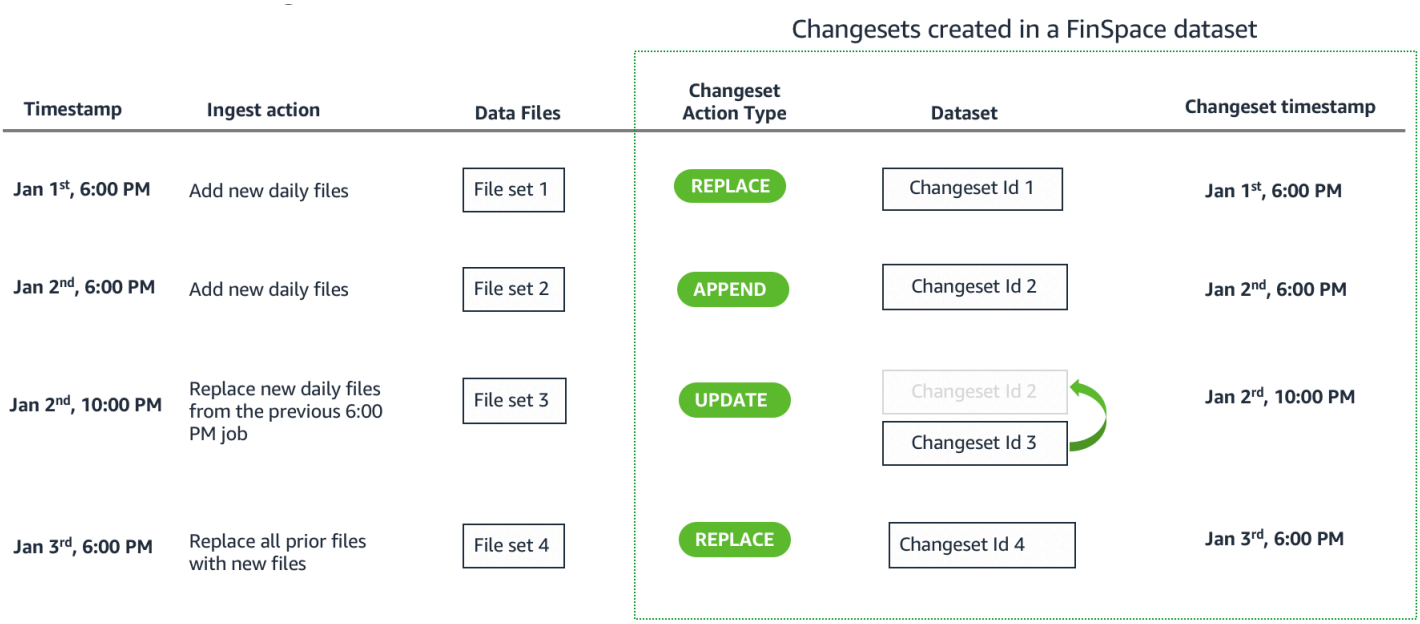

You can ingest data into Amazon FinSpace from your enterprise data lake or on-premises data stores. The data is ingested into Datasets. FinSpace supports ingestion of structured data such as CSV, parquet, XML, and JSON or any unstructured data files. You can ingest data using the FinSpace web application or the SDK. To learn more about loading data, see Adding and managing data in Amazon FinSpace.

### Datasets

Dataset is a logical container of semantically identical data and schema. Data is ingested into a dataset as a changeset, and every time a new set of data is added to a dataset, a changeset is created. Dataset tracks the versions of data that is ingested as changesets. Data Views are generated from the changesets which can be analyzed within the FinSpace Notebook

environments. A typical FinSpace environment may contain hundreds or thousands of datasets. To learn more, see Working with datasets.

## Changesets




A changeset is created when a new set of data files are ingested in a dataset in a single ingest operation. For example, if a data source sends files at the end of the day everyday for a data product, you can create a new changeset by ingesting the files. A changeset is created with a unique id and a timestamp for data versioning. You can create changesets to add new data, replace previously added data, and also make corrections to specific changesets. To learn more, see changesets.

# Data organization

Datasets can be described, organized, and made browsable and searchable in FinSpace. You can build a business data catalog with business terms and taxonomy specific to your organization. The organizational concepts provided in Finspace are designed to provide centralized governance and control. The cataloging structure needs to be defined once with definition of meta data fields. The permissions to define the catalog and metadata fields can be restricted to data governor or data stewards. Once the cataloging structure is defined, the metadata fields can be associated to data to automatically organize it.

## Categories

Categories allow for cataloging of datasets by commonly used business terms. Categories are hierarchical in nature, allowing for each node of the hierarchy to have a name and a description. The order of the nodes within a level are defined when you define categories. The categories are displayed in the data browser on the left side of the FinSpace web application home page. The FinSpace users will use the data browser to browse datasets.

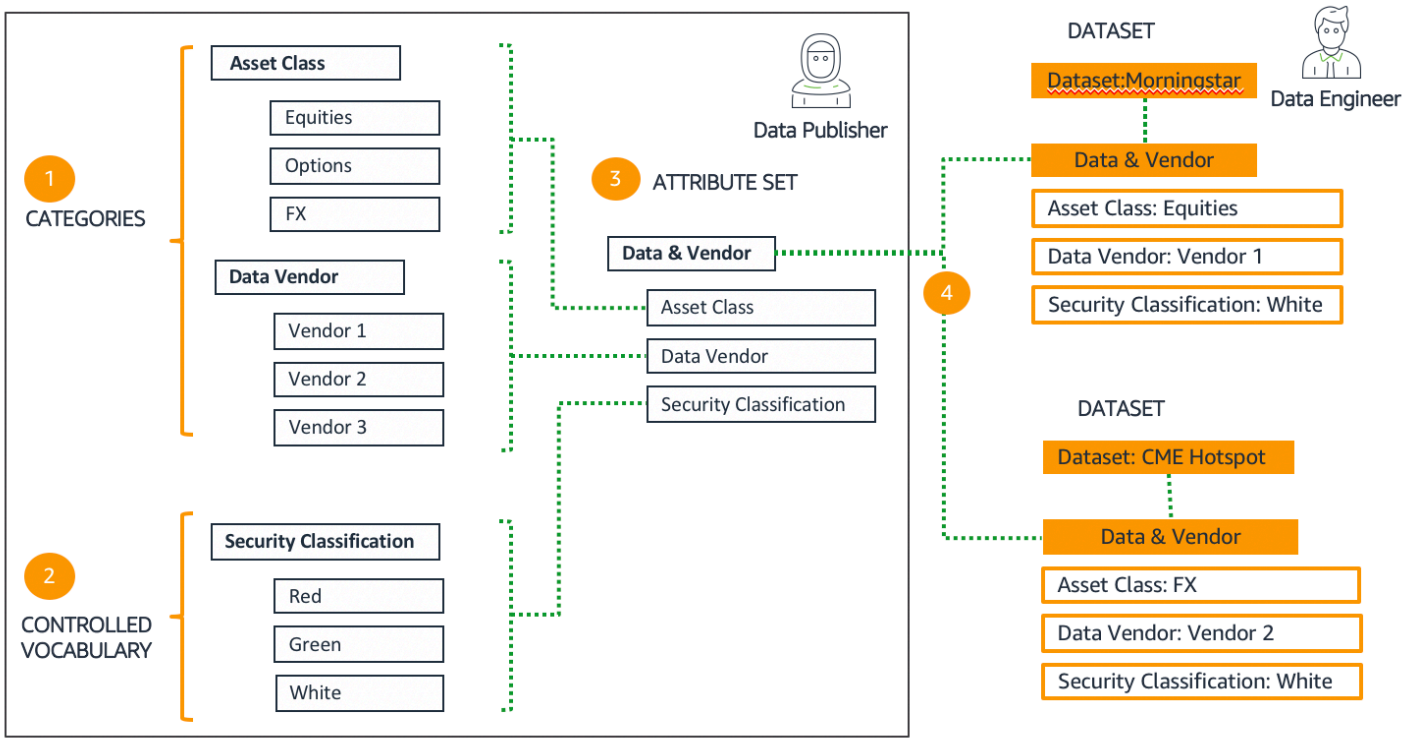
## Controlled vocabularies

Controlled vocabularies are enumeration lists of attributes to describe datasets. A controlled vocabulary is used to ensure that standardized terms are used to describe a dataset. For example, if your organization has a data security classification scheme with terms such as `Red`, `Green`, `White` to describe the data, you can create a controlled vocabulary with the name Security Classification, with values `Red`, `Green`, `White`. The controlled vocabulary, `Security Classification`, can then be used as an attribute field to describe a dataset where only one of three values (`Red`, `Green`, `White`) can be applied.

## Attribute sets

Attribute sets are lists of attributes that can be applied to describe datasets. Attributes are metadata fields used to capture additional business context for each dataset. Attribute sets help you ensure the consistent capture of metadata which increases metadata quality and provides better search results for users. You can then browse and search attributes to find a dataset based on the values assigned to the attributes.

You can configure a business data catalog with above concepts in FinSpace in four steps.




1. Build categories – In the first step, you define the categories and sub-categories with business terms. The categories are displayed in the data Browser on the left side of the FinSpace web application home page. The data browser is one of the two ways for a user to search for data; the other way is the search bar.
2. Build controlled vocabularies – In this step, you define the controlled vocabularies to use in your organization. Example of a controlled vocabulary is data sensitivity classification.
3. Define attribute sets – In this step, you define the attribute sets. You can define an attribute type of a pre-defined category or controlled vocabulary.
4. Associate attribute set with a dataset – Once an attribute set is defined, it can be associated with a dataset. The dataset is then described by setting the values of the attributes.

A data governor or data steward can define categories, controlled vocabularies, and attribute sets, and data engineers can associate attribute sets with datasets. Step 1, 2, 3 are one-time actions, the categories and controlled vocabularies can be reused in defining new attribute sets, and an attribute set can be associated with multiple datasets. To learn more, install the capital markets sample data bundle, which creates a business data catalog with example categories, controlled vocabularies, and attribute sets that are associated with the provided sample datasets.

## Data views

Data views provide access to the data stored in a dataset. A data view represents the full picture of a dataset at a given point of time. Data views create an optimized input data structure for efficient querying of data. Multiple data views can be created from a dataset that cover different time periods. The data management engine in FinSpace supports *bi-temporality* that allows you to create a view of the data as-of a particular point in time, factoring in or eliminating corrections to data. Bi-temporality enables you to reproduce the results as if they were calculated with a version of data on a past date. In addition, the results of an analysis such as the output dataset and parameters can be stored in a separate FinSpace data set for future reproducibility.

## Data preparation and analysis

You can load data views in the FinSpace notebook environment to prepare and analyze data at petabytes scale. To learn more, see Prepare and analyze data in Amazon FinSpace.

### Jupyter lab notebook

The notebook environment in FinSpace supports Jupyter Lab notebooks for writing code to analyze the data. You can access the datasets created in FinSpace from the notebooks using the APIs and load the data views and run analysis. To learn more, see Working with notebooks.

### Managed Apache spark clusters

FinSpace supports managed Spark clusters that can be instantiated from the notebooks with API calls. The Spark clusters allow parallelization of data analysis and available in five sizes. To learn more, see Working with Spark clusters.

### Time series library

FinSpace provides a time series analytics library to prepare and analyze historical financial time series data using FinSpace managed Spark clusters. You can use the library to analyze high-density data like US options historical OPRA (Options Price Reporting Authority) with billions of daily events or sparse time series data such as quotes for fixed income instruments. To learn more, see Spark time series analytics.

## Integrated permission management

FinSpace supports an integrated security and governance model. Users of FinSpace are registered in FinSpace and assigned permissions on an FinSpace application and dataset level. The same

permissions are applied to results you see in the catalog and data you can access in the notebook and APIs. To learn more, see Managing user access in Amazon FinSpace.

### Superuser

A superuser has all the permissions in FinSpace. The first superuser for your FinSpace environment is created from the AWS console. The superuser can then create other superusers and application users from the FinSpace web application. We recommend that you only use the superuser for the initial setup, and use application users with assigned permissions for regular application access.

### Application user

An application user does not have any permissions when their account is created. They are assigned permissions by adding them to a permission group.

### Permission groups

Permission groups contain users. Permissions to perform any action in FinSpace are assigned to permission groups, not directly to the user. A user can be a member of multiple permission groups. A permission group cannot be a member of another permission group.

### Permissions

Permissions are assigned to permission groups and not to users. The are two kinds of permissions in FinSpace - application permissions and dataset permissions. Application permissions are assigned to a permission group when creating or editing it (for example, create datasets). Dataset permissions are assigned on a per dataset basis when associating a permission group to a dataset (for example, read a view in a dataset).

## Audit report

From the FinSpace web application, you can generate audit reports to support your compliance processes. FinSpace tracks all activity within a FinSpace environment. You can restrict access to audit reports.

# Managed kdb Insights

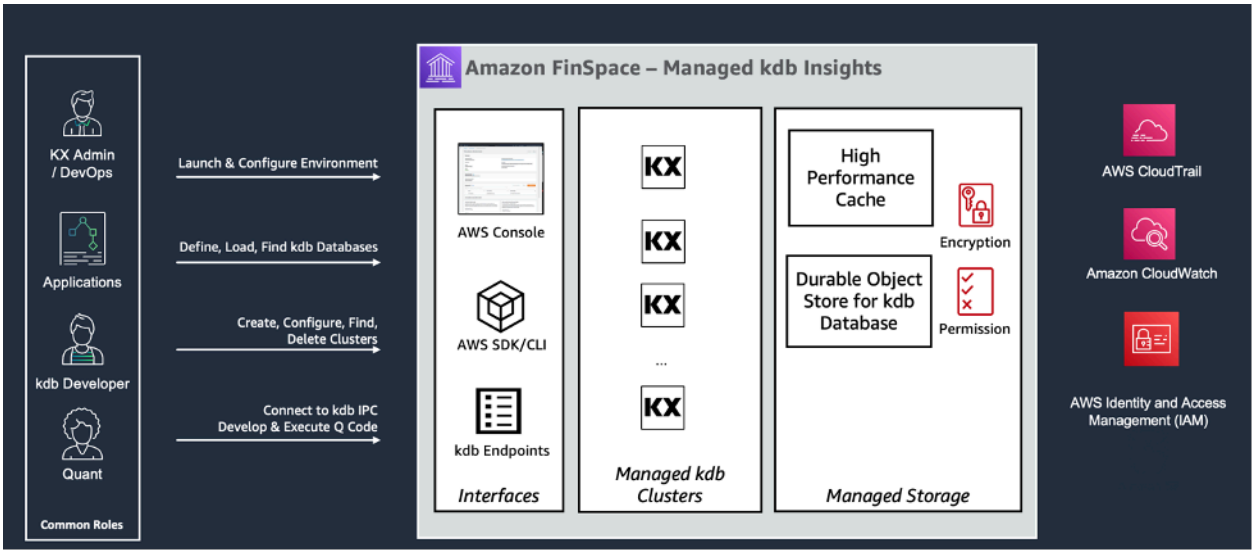
Amazon FinSpace with Managed kdb Insights provides a fully managed service for the latest version of kdb's analytics engine. Kdb Insights is the leading time-series analytics engine used by capital markets customers to power their business-critical analytics workloads, such as liquidity insights, pricing, transaction cost analysis, and back-testing.

With Amazon FinSpace Managed kdb Insights, you can deploy your existing real-time stream processing and high-performance kdb code in the AWS cloud to power time-sensitive analytics workloads. This is fundamental to running investment and trading businesses.

Using Managed kdb Insights' clusters, you can quickly set up a managed data processing and analytics hub. With a few clicks in the Managed kdb Insights application, you can migrate your existing kdb datasets to FinSpace. By configuring Managed kdb Insights to auto scale the kdb clusters up and down, you can meet the availability and runtime performance needs of each of your kdb workloads. You can configure your Managed kdb Insights clusters to automatically deploy across multiple Availability Zones and Regions to ensure that your analytics environment is available during the most critical business hours.

As a result, you no longer require teams of specialists to monitor the infrastructure. This is because Managed kdb Insights continuously monitors underlying server health and capacity, and automatically replaces servers when they fail and patches servers in need of updates. In addition, it simplifies the work required to set up and deploy new clusters so the kdb administrators can focus more on business needs. Managed kdb Insights also supports running the same customer-developed kdb scripts that they run today on premises, and provides the same familiar kdb interfaces.

**How it works**




**The diagram describes key components of Managed kdb Insights:**

- You can manage Managed kdb Insights resources by using the AWS Management Console or the SDK/CLI.
- Data is stored in durable object store back databases.
- Compute clusters running kdb software access data in the database.
- Data can be cached from the database on high performance disk cache for fast access by the cluster.
- Developers and quantitative analysts (quants) can access clusters via kdb IPC connections.
- Access can be controlled through IAM.
- Activity is logged to CloudTrail and CloudWatch.

**Topics**

# Permissions required for Managed kdb

You must have certain IAM permissions to use Managed kdb. In addition to the finspace:*permissions, you might need additional permissions to use the resources in your AWS account. FinSpace uses these permissions on your behalf to configure resources in your account where it needs to function. Add these permissions by using the IAM policies to IAM roles that you use to interact with Managed kdb.

The following table shows a list of permissions and what they are needed for.

| Permissions (IAM actions) | Use for | Used by |
|---|---|---|
| "logs:CreateLogDelivery" | Creating and deleting CloudWatch logs | Users who create or delete the clusters |
| "logs:GetLogDelivery" | | |
| "logs:UpdateLogDelivery" | | |
| "logs:DeleteLogDelivery" | | |
| "logs:ListLogDeliveries" | | |
| "logs:PutResourcePolicy" | | |
| "logs:DescribeResourcePolicies" | | |
| "logs:DescribeLogGroup" | | |

| Permissions (IAM actions) | Use for | Used by |
| --- | --- | --- |
| "logs:CreateLogGroup" | | |
| "ec2:CreateVpcEndpoint" | Managing kdb clusters | Users who create or delete the clusters |
| "ec2:DeleteVpcEndpoints" | | |
| "ec2:DescribeSubnets" | | |
| "ec2:AcceptTransitGatewayVpcAttachment" | Creating a connection between your Managed kdb environment and your transit gateway | Administrators who configure the transit gateway environment using the `UpdateKxEnvironmentNetwork` API |
| "ec2:DescribeSubnets" | | |
| "ram:CreateResourceShare" | Creating a resource share on the transit gateway | Users who update kdb environment |
| “ram:GetResourceShareInvitiations” | Accepting resource share on private certificate authority for cluster TLS connection | Users who create kdb environment |
| "ram:AcceptResourceShareInvitation" | | |
| "iam:CreateServiceLinkedRole" | Creating the FinSpace service-linked role (SLR) when creating a kdb environment | Users who create kdb environment |

| Permissions (IAM actions) | Use for | Used by |
|---|---|---|
| "ec2:DescribeTags" | Creating and describing tags on FinSpace managed VPC endpoints | Users who create and delete clusters |
| "ec2:CreateTags" | | |
| "finspace:*" | Performing actions to manage FinSpace resources | Users that manage resources in FinSpace |
| "kms:CreateGrant" | Encrypting any customer data at rest | Users who create kdb environment |
| "kms:ReireGrant" | | |
| "ec2:DescribeTransitGateways" | Checking if the transit gateway exists | Users who configure the transit gateway environment using the UpdateKxEnvironmentNetwork API |
| "s3:GetObject" | Controlling access for ingesting code and data into the service. | Users who create clusters, update code on clusters, or create changesets. See the sections below for additional details. |
| "s3:GetObjectTagging" | | |
| "s3:GetObjectVersion" | | |
| "s3:ListBucket" | | |

## Permissions FinSpace needs to resources in your account

You will need to grant permission to FinSpace to access certain resources in your account. To do this, follow steps in the following sections.

## Granting permission to your AWS KMS key to encrypt data and code stored in Managed kdb

You must grant the FinSpace service access by using the AWS KMS key policy to create Managed kdb changesets and load code onto a cluster. The following is an example of such a policy.

In the following example, replace each *user input placeholder* with your own values.

### Sample AWS KMS key policy

```
{
    "Version": "2012-10-17",
    "Id": "FinSpaceServiceAccess",
    "Statement": [{
            "Sid": "FinSpace Permissions",
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "arn:aws:kms:us-east-1:555555555555:key/f935d84c-
d365-4753-875Y-1c014ab4f61Z",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                }
            }
        }

    ]
}
```

## Granting permission to your Amazon S3 code bucket to load code onto your Managed kdb cluster

To load code onto your cluster you must first grant the FinSpace service access to the Amazon S3 bucket that stores the code you want to load. The following is an example of the policy that you can use to grant access to code location.

### Example policy to grant access to the code location

In the following example, replace each *user input placeholder* with your own values.

```
{
    "Version": "2012-10-17",
    "Id": "FinSpaceServiceAccess",
    "Statement": [{
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::managed-kdb-code/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:finspace:us-
east-1:555555555555:kxEnvironment/<EnvironmentID>/*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::managed-kdb-code",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                },
                  "ArnLike": {
                    "aws:SourceArn": "arn:aws:finspace:us-
east-1:555555555555:kxEnvironment/<EnvironmentID>/*"
                }
            }
        }
    ]
}
```

After you grant the FinSpace service access to the S3 bucket , you must ensure that the IAM role that you use when you create a cluster or when you update the code on a cluster has permission to access the files on the Amazon S3 bucket. The following is an example of the policy that you can use to grant access to the role.

In the following example, replace each *user input placeholder* with your own values.

**Example policy for granting calling role access to the code location**

```
{
    "Version": "2012-10-17",
    "Id": "FinSpaceServiceAccess",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::managed-kdb-code/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::managed-kdb-code"
        }
    ]
}
```

When you set permissions on the role, you can control which Amazon S3 locations a user can access. You can also set *Deny* policies on this role to prevent access to resources. For example, you can use the *Deny* policy to prevent access to resources in another account.

## Granting permission to your Amazon S3 data staging bucket to ingest data into Managed kdb

To ingest data from Amazon S3 into your database through a changeset, you must first grant FinSpace access to the S3 bucket that stores the data you want to import as Managed kdb changesets. The following is an example of such a policy.

In the following example, replace each *user input placeholder* with your own values.

### Example policy to grant the FinSpace service principal access to the code location

In the following example, replace each *user input placeholder* with your own values.

```
{
    "Version": "2012-10-17",
    "Id": "FinSpaceServiceAccess",
    "Statement": [{
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::managed-kdb-data/*",

            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:finspace:us-
east-1:555555555555:kxEnvironment/<EnvironmentID>/*"
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::managed-kdb-data",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:finspace:us-
east-1:555555555555:kxEnvironment/<EnvironmentID>/*"
                }
            }
        }
    ]
}
```

After you grant FinSpace access to the Amazon S3 bucket, you must ensure that the IAM role you use when you create a changeset has permission to access the files on the Amazon S3 bucket. The following is an example of such a policy.

In the following example, replace each *user input placeholder* with your own values.

**Example policy to grant role access to the changeset location**

```
{
    "Version": "2012-10-17",
    "Id": "FinSpaceServiceAccess",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::managed-kdb-data/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
```

```
            "Resource": "arn:aws:s3:::managed-kdb-data"
        }
    ]
}
```

When you set permissions on the role, you can control which Amazon S3 locations a user can access. You can also set *Deny* policies on this role to prevent access to resources. For example, you can use the *Deny* policy to prevent access to resources in another account.

# Managed kdb Insights environments

The Managed kdb Insights environment provides a logical container where you can launch and run clusters, and store data from kdb that can be used by the clusters.

All resources in the Managed kdb environment run in AWS managed accounts and not in the customer account. The Managed kdb environment dedicated account is not shared with the existing FinSpace dataset browser environment.

## Managing kdb environments

The following sections provide a detailed overview of the operations that you can perform by using a Managed kdb Insights environment.

### Creating a kdb environment

> **Note**
>
> You can only create one kdb environment per Region per AWS account.

**To create a kdb environment**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. On the getting started page, choose **Create kdb environment**.
4. On **Create kdb environment** page, enter the environment name and description.

5. Choose a symmetric encryption KMS key to encrypt data in your kdb environment. If a KMS key is not available in the Region where you want to create your FinSpace environment, create a new key.

   For more information, see Creating keys in the *AWS Key Management Service Developer Guide*.

6. (Optional) Add a new tag to assign it to your kdb environment. For more information, see AWS tags.

   > **Note**
   >
   > You can only add up to 50 tags to your environment.

7. Choose **Create kdb environment**. The environment creation process begins and the environment details page opens. The environment creation process takes few minutes to finish in the background.

   You can view the status of environment creation under the kdb environment configuration section.

   After the environment is successfully created, you can add network configuration, databases, and clusters to the environment.

## Updating a kdb environment

### To update a kdb environment

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Edit**.
5. Edit the environment details.

   > **Note**
   >
   > You can only edit the **Name** and **Description** .

6. Choose **Update kdb environment**. You can view the updated details on the environment details page.

## Viewing kdb environment details

### To view and get details of a kdb environment

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.

   The environment details page opens where you can view details about the environment, add or view network configuration, create new databases, and add clusters.

## Deleting a kdb environment

> **Note**
>
> This action is irreversible. Deleting a kdb environment will delete all resources (users, clusters, and databases) and their metadata in the account. After you initiate a deletion request, the billing for resources in an environment will stop immediately.

### To delete a kdb environment

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Delete**.
5. On the confirmation dialog box, enter *confirm*.
6. Choose **Delete**.

# Managing environment network settings

For each Managed kdb Insights environment, you can configure a network connection to allow the Managed kdb clusters running in your environment infrastructure account to access resources in your internal network. You can create a connection by connecting your infrastructure account to an existing transit gateway in your organization.

After you add a network, you can also specify details for the DNS servers that your Managed kdb clusters will use to resolve resources outside of your Managed kdb environment. After your Managed kdb environment is connected to your network, you can optionally configure your network to allow outbound traffic from your environment to the internet. This connectivity is managed by your network infrastructure. Managed kdb doesn't support direct internet access (inbound or outbound).

## Prerequisites

Before you proceed, complete the following prerequisites:

- Make sure that a kdb environment has been created. For more information, see Creating a kdb environment.
- Make sure that a transit gateway has been created in AWS Transit Gateway. For more information, see Create the transit gateway in the *AWS Transit Gateway User Guide*.
- Make sure that you have a */26 (64)* IP address range from the *100.64.0.0/10* range that you can allocate to the subnets that connect to your transit gateway.

## Creating a network connection

You can configure a network connection to allow the Managed kdb clusters running in your environment infrastructure account to access resources in your internal network.

Optionally, you can also define how you manage the outbound traffic from kdb network to your internal network. You do this by configuring the attachment network access control lists (ACLs).

A network ACL allows or denies specific outbound traffic at the subnet level. You can use the default network ACL for your VPC. Alternatively, to add an additional layer of security to your VPC, you can create a custom network ACL for your VPC with rules that are similar to the rules for your security groups. For more information, see the Network ACL rules in the *Amazon VPC User Guide*.

**Note**

- You can only configure one network connection per Managed kdb environment.
- You cannot delete a network connection. To remove the existing network and the network ACL attachments, delete the Managed kdb environment.

**To create a network connection**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, under **Network** tab, choose **Add network configuration**.
5. On **Add network configuration** page, enter a transit gateway ID and the CIDR range that will be used for the subnets connecting to your internal network. For more information, see the *Amazon VPC Transit Gateways User Guide.*

   **Note**

   When you add a transit gateway without creating a network ACL, all outbound traffic is allowed by default.

6. (Optional) Add rules to define how you want to manage the outbound traffic from kdb network to your internal network. Choose **Add new rule** to allow or deny outbound traffic for each port range and destination.

   **Note**

   - When you create a network ACL rule, by default all the other traffic are denied.
   - We process the ACL rules according to the rule numbers, in ascending order.

7. Choose **Save**. The connection creation process begins and the environment details page opens from where you can check the status under the **Network** tab.

## Editing a network

> **Note**
>
> - You can't edit the transit gateway ID and CIDR routable space for your network.
> - You only edit the network ACL configurations for your network.

### To edit a network connection

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, under **Network** tab, choose **Edit network**.
5. On **Edit network** page, add or modify the network ACL rules as required.
6. Choose **Save changes**. The updates are available on the environment details page.

## Adding DNS details

You can set the DNS resolver that the Managed kdb Insights compute nodes will use for resolving IP addresses. This is useful if you want to connect from your Managed kdb compute clusters to resources like on-premises kdb ticker plants or other resources. We recommend you add DNS details only after you have successfully configured a network in your Managed kdb environment.

> **Note**
>
> You can only add one DNS server and IP address per Managed kdb environment.

### To add DNS details

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.

3. From the kdb environments table, choose the name of the environment.
4. Under **DNS details**, choose **Add details**.
5. On **Add DNS details** page, enter the DNS server name and IP address that the clusters running in the Managed kdb environment will use.
6. Choose **Add DNS details**. The **environment details** page opens and the DNS details are added in the **DNS details** section, from where you can edit the DNS details.

# Tutorial: Configuring and validating outbound network connectivity through transit gateway

Amazon FinSpace Managed kdb environment allows you to connect to kdb or q processes in your account through transit gateway, without going over the internet. This section demonstrates how to setup outbound network connectivity from FinSpace Managed kdb environment to your virtual private cloud (VPC) and validate connectivity from an RDB cluster to a q process on an Amazon EC2 instance in your network.

**Topics**



## Prerequisites

Before you proceed, complete the following prerequisites:

- Create a kdb environment. For more information, see Creating a kdb environment.

> **Note**
>
> Note down the `Availability Zone Ids` after creating a kdb environment. You will need them when you create an attachment from your VPC to a transit gateway.

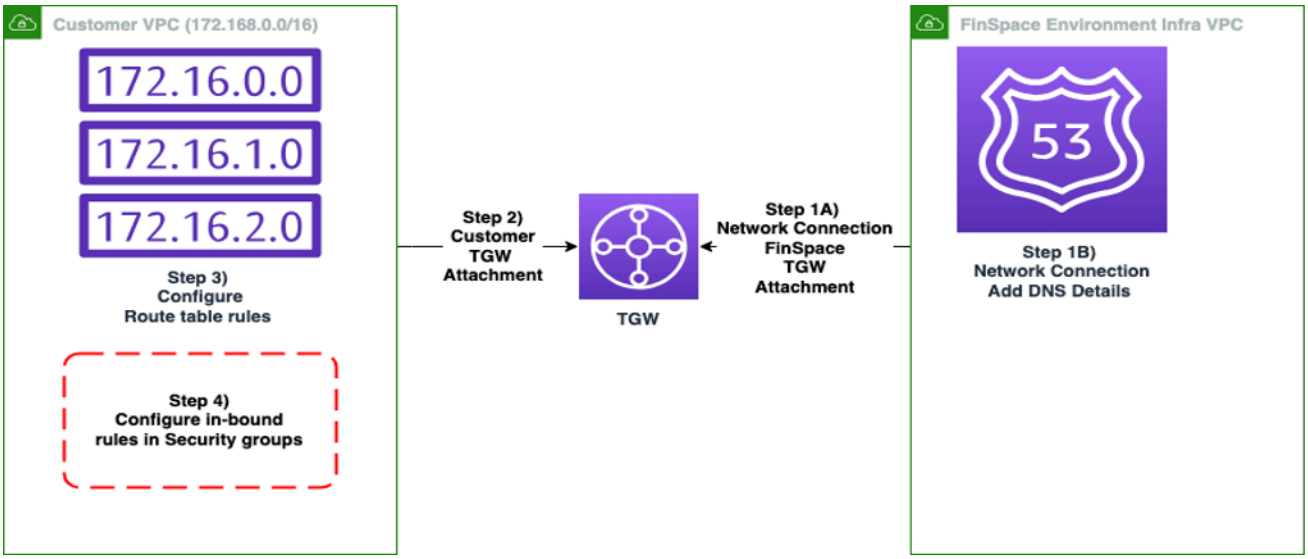
- Make sure that you create a transit gateway in AWS Transit Gateway. For more information, see Creating the transit gateway in the *Amazon VPC Transit Gateways User Guide*.

> **Note**
>
> When creating the transit gateway, you only need to specify the name and description. For the rest of the fields, choose the default values. For example, for DNS-Support, VPN ECMP support, Default route table association, and Default route table propagation options should be selected by default.

- Make sure you are familiar with the process of the section called "Creating a kdb environment", the section called "Creating a kdb user", and the section called "Creating a cluster".

## Setup diagram

This diagram shows a high level of configuration steps that are further described in the following sections.



## Step 1: Configuring a network connection to create FinSpace VPC transit gateway attachment

### To create a network connection

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, under **Network** tab, choose **Add network configuration**.
5. On **Add network configuration** page, enter a transit gateway ID and the CIDR range that will be used for the subnets connecting to your internal network. For more information, see the *Amazon VPC Transit Gateways User Guide.*

   > **Note**
   >
   > When you add a transit gateway without creating a network ACL, all outbound traffic is allowed by default.

6. (Optional) Add rules to define how you want to manage the outbound traffic from kdb network to your internal network. Choose **Add new rule** to allow or deny outbound traffic for each port range and destination.

   > **Note**
   >
   > - When you create a network ACL rule, by default all the other traffic are denied.
   > - We process the ACL rules according to the rule numbers, in ascending order.

7. Choose **Save**. The connection creation process begins and the environment details page opens from where you can check the status under the **Network** tab.

> **Note**
>
> - When you configure a network connection, make sure that you have a /26 (64) IP address range from the *100.64.0.0/10* range. The CIDR range should not be used in your network

or any other environments that are connected by this TGW. A few valid examples of this CIDR range are *100.64.0.0/26*, *100.64.1.0/26*, *100.64.2.0/26*, *100.64.3.0/26*. We will pick *100.64.0.0/26* for this tutorial.

- This step creates a transit gateway VPC attachment to connect FinSpace environment to the transit gateway. After you configure a network, check the **Network** tab for details of your network.

## Step 2: Adding DNS details to your network connection

The **Network** tab on the Kdb environments details page allows you to add custom DNS server name and IP address. This is used when you have a custom DNS server that you want to query for internal host names. The DNS server IP is used for DNS resolution of queries.

**To add DNS details**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. Under **DNS details**, choose **Add details**.
5. On **Add DNS details** page, enter *example.com* as the DNS server name and *173.31.0.2* as the DNS server IP. This means that any DNS queries for *example.com* from the FinSpace clusters will return the DNS resolver at *172.31.0.2* in the your VPC.

   > **Note**
   >
   > The IP *172.31.0.2* is the second IP address in the default VPC CIDR and corresponds to the IP of the DNS Resolver for an Amazon VPC. Any DNS queries for *example.com* from the FinSpace clusters will return the DNS resolver at *172.31.0.2* in your custom VPC.

6. Choose **Add DNS details**. The **environment details** page opens and the DNS details are added in the **DNS details** section, from where you can edit the DNS details.

## Step 3: Setting up a transit gateway VPC attachment from your VPC

> **Note**
>
> It may take a few minutes for Step 1 and Step 2 to complete. Wait till these steps are successful before proceeding.

In the previous step you created a network connectivity from FinSpace environment to your transit gateway but FinSpace cannot reach into your network unless you create a VPC attachment from your VPC to Transit Gateway and set up routing and rules for the traffic to flow into your network.

In this step, you create a transit gateway attachment and validate that it is associated in the transit gateway associations.

**To create a transit gateway VPC attachment from your VPC**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Choose **Create transit gateway attachment**.
4. For **Transit gateway ID**, choose the transit gateway for the attachment that you created in step 1 of this tutorial.
5. For **Attachment type**, choose **VPC**.
6. For **VPC ID**, choose the default VPC to attach to the transit gateway. This VPC must have at least one subnet associated with it.

   > **Note**
   >
   > There is a default VPC for every AWS account. The default VPC ID is the value of the VPC ID column of the VPC table. To view your default VPC:
   >
   > 1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
   > 2. In the navigation pane, choose **Your VPCs**.
   > 3. In the **Default VPC** column, look for a value of **Yes**. Take note of the ID of the default VPC.

7. For **Subnet IDs**, choose 3 subnets from the availability zones where the environment is created.

   To check the availability zones ID mapping for your AWS account, go to the AWS Resource Access Manager in your account. Navigate to the product console, find the AZ ID at the bottom right of the page.

**To validate the TGW associations**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Transit Gateway ID** for transit gateway that you created earlier.
3. Under **Details**, choose **Association route table ID**. The **Association** tab shows the two VPC attachments, one from FinSpace infrastructure VPC and the other from your VPC.

## Step 4: Configuring routes in your VPC route tables

With a VPC, you must create routes to send traffic to the transit gateway. The following steps show how you can update your default VPC route tables to have an entry for traffic to return to FinSpace VPC.

**To configure route tables**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the navigation pane, choose **Route Tables**.
3. Choose the route table for the default VPC ID.
4. Choose **Edit routes**.
5. On **Edit routes** page, choose **Add route** and enter *100.64.0.0/26* as the **Destination**. This value is the same as the CIDR range that you added while creating the network connectivity in the section called “Step 1: Configuring a network connection”.
6. For **Target** choose **Transit Gateway** and select your transit gateway ID.
7. Choose **Save changes**.

## Step 5: Configuring security group inbound rules

After you set up routing, you need to add inbound rule for the default security group to allow inbound traffic. The default security group comes with your AWS account. For more information, see Default security groups in the *Amazon VPC User Guide*.

A security group acts as a firewall that controls the traffic allowed to and from the resources in your VPC. You can choose the ports and protocols to allow for inbound traffic or outbound traffic. For each security group, you add separate sets of rules for inbound traffic and outbound traffic. For more information, see Security group rules in the *Amazon VPC User Guide*.

As an example, add an entry to allow TCP traffic for port *5005* to connect to a q process in your account running on port *5005*. This makes port *5005* of any host launched with the default security group to be reachable.

**To create an inbound rule**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose **Security Groups**.
3. Under the **Inbound rules** tab, choose **Edit inbound rules**.
4. On **Inbound rules** page, choose **Add rules**.
5. For **Type**, choose *Custom TCP*.
6. For **Port range** enter *5005*.

   As another example, you can also allow all traffic from FinSpace to all ports. To allow all ports by default, follow the above steps of creating an inbound rule. In step 5, for **Type**, choose *All TCP*.

   **Note**

   - If you need to restrict outbound traffic to specific ports and destination, add network ACL while creating a network connection to deny outbound traffic from FinSpace for each port range and destination.
   - When you create an Amazon EC2 instance, you need to specify the default security group for these inbound rules to apply. See next section for an example of how an Amazon EC2 instance is created with this security group.

If you have hosts with different port rules you can create a security group for each host. When you launch an EC2 instance, use the security group with the port rules for your host.

## Step 6: Validating network connectivity

After you've successfully created an outbound network connectivity between FinSpace VPC and your VPC using transit gateway, you can validate the network configuration. To do this, run a test to connect to a customer EC2 instance q process from an RDB cluster in the FinSpace environment.

The following procedure shows how to connect to an RDB cluster and then connect to a q/kdb process running on EC2 instance in the your VPC account. In this step, you will create two EC2 instances:

- **customerEc2Instance** – This is a q process to which the RDB would connect to.
- **clientEc2Instance** – This is a q client to connect to the RDB cluster.

### Create an RDB Cluster

Create an RDB cluster with a single-AZ mode by following the steps in this tutorial.

### Create an EC2 instance

Use the following command to create an EC2 instance with a name *customerEc2Instance* instance to which an RDB would connect to.

```
echo '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"Service":"ec2.amazonaws.com"},"Action":"sts:AssumeRole"}]}' > policy.json
aws iam create-role --role-name ssmrole --assume-role-policy-document file://
policy.json
aws iam attach-role-policy --role-name ssmrole --policy-arn arn:aws:iam::aws:policy/
AmazonEC2ContainerRegistryFullAccess
aws iam attach-role-policy --role-name ssmrole --policy-arn arn:aws:iam::aws:policy/
AmazonSSMManagedInstanceCore
aws iam attach-role-policy --role-name ssmrole --policy-arn arn:aws:iam::aws:policy/
AmazonSSMPatchAssociation
aws iam create-instance-profile --instance-profile-name "SSMRole"
aws iam add-role-to-instance-profile --instance-profile-name SSMRole --role-name
 ssmrole
```

```
aws ec2 run-instances \
--count 1 \
--instance-type t2.micro \
--security-group-ids <SecurityGroup>\
--subnet-id <SUBNET> \
--iam-instance-profile Name=SSMRole \
--tag-specifications
 "ResourceType=instance,Tags=[{Key=Name,Value=CustomerEc2Instance}]" \
--image-id $(aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-
ami-hvm-x86_64-gp2 --region us-east-2 | jq ".Parameters[0].Value" -r) \
--metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

### Start a q process and listen on port 5005

1. Connect to the *CustomerEc2Instance* instance. For more information, see this section.
2. Install the q client. For more information on installation, see Installing kdb+.
3. Launch a q process and run the following command to listen on port *5005*.

```
q) \p 5005
```

### Create another EC2 instance

Create another instance with a name *clientEc2Instance*, which you can use to connect to the RDB cluster. The EC2 instance should use the same security group and subnet that you chose for the cluster.

```
aws ec2 run-instances \
--count 1 \
--instance-type t2.micro \
--security-group-ids <security group> \
--subnet-id <SUBNET> \
--iam-instance-profile Name=SSMRole \
--tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=Bastion}]" \
--image-id $(aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-
ami-hvm-x86_64-gp2 --region us-east-1 | jq ".Parameters[0].Value" -r) \
--metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

**Test the connection**

Test the connection from q process on EC2 instance to the RDB cluster.

Create an RDB cluster with a single-AZ mode by following the steps in this tutorial.

1. Connect to the *clientEc2Instance* by following the steps in this section.
2. Install the q client. For more information on installation, see Installing kdb+.
3. Start a q process and connect to the RDB cluster on port *5005* by using the following example command.

```
q)cs_rdb1: <RDB cluster connection string>
           q)cs_rdb1: ssr[cs_rdb1;"\n";""]
           q)conn: hopen cs_rdb1
           q)conn hopen(":<Private IP DNS name
           of customerEc2Instance
           5005"; 10)
```

   The following section explains the sample code:

   - *cs_rdb1* has a cluster connection string. For more information on how to get a connection string, see the Interacting with a kdb cluster section.
   - *hopen* command opens a connection to the RDB cluster and gets a connection handle.
   - Use connection handle to run *hopen* connection test to the *customerEc2Instance* q process listening on port *5005* to test connectivity from RDB cluster to *customerEc2Instance*.

You should be able to successfully connect to port *5005*.

Repeat the steps for starting a q process and testing connection with port *5006*. You will fail to connect because only port *5005* is allowed in the in-bound rules of the security groups.

## Step 7: Validating connection using the DNS server configuration

As an example, create a private hosted zone in your account that has an A record rule for *example.com* and Private IP DNS name of *customerEc2Instance*.

To create a private hosted zone, see Creating a private hosted zone in the *Amazon Route 53 User Guide*. To create a record rule, see this section.

Start a q process and connect to the RDB cluster on port *5005* by using the following example command.

```
q)cs_rdb1: <RDB cluster connection string>
q)cs_rdb1: ssr[cs_rdb1;"\n";""]
q)conn: hopen cs_rdb1
q)conn hopen(":<Private IP DNS name of customerEc2Instance 5005"; 10)
```

Next, run the following command to test connection on port *5005* by using the DNS name *example.com*.

```
q)cs_rdb1: <RDB cluster connection string>
q)cs_rdb1: ssr[cs_rdb1;"\n";""]
q)conn: hopen cs_rdb1
q)conn hopen(":example.com:5005"; 10)
```

The connection test using the DNS name should work successfully.

# Managed kdb Insights databases

A Managed kdb Insights database acts as a highly available and scalable repository to store your kdb data files so that they can be used with one or more historical database (HDB) clusters in FinSpace kdb. Data in a database may consist of either kdb objects, kdb splayed tables, or kdb partitioned tables. These represent different types of kdb table structures and each must follow a prescribed file and path layout. You can learn more about each of these structures here.

Data is loaded into a database by defining a changeset, which lets you import a file or set of files into a database. The files in the kdb database are placed into logical paths called the *Database paths*. Creating a database does not automatically load any data. You must add data to the kdb database through changesets.

## Managing kdb databases

The following sections provide a detailed overview of the operations that you can perform by using a Managed kdb database.

### Creating a kdb database

**To create a kdb database**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Databases** tab.
5. Choose **Create database**.
6. On the **Create database** page, enter a unique name for the database.
7. (Optional) Enter a description for your database.
8. (Optional) Add a new tag to assign it to your kdb database. For more information, see AWS tags.

   > **Note**
   >
   > You can only add up to 50 tags to your database.

9. Choose **Create database**. The environment details page opens and the table under **Databases** lists the newly created database.

   You can choose the database name from the list to view its details in database details page.

## Managing data in a kdb database

The Managed kdb Insights database allows you to add, update, or delete a set of files. When you create a database, there is no data loaded in it. You must add data to the database through changesets. A changeset represents a versioned set of changes that are applied to a database.

### Creating an Amazon S3 bucket policy

Before you can ingest data into your database, you must have a valid Amazon S3 bucket IAM policy in place to allow FinSpace to access the data you will ingest into it. The following is an example of such a policy.

In the following example, replace each *`user input placeholder`* with your own values. Replace *`555555555555`* with the AWS account where you created your Managed kdb Insights environment.

### Example — Sample Amazon S3 bucket policy

```
{
    "Version": "2012-10-17",
    "Id": "FinSpaceServiceAccess",
    "Statement": [{
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:::managed-kdb-data/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::managed-kdb-data",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555"
                }
            }
        }
    ]
}
```

### Creating a new changeset

You can add, update, and delete data in a database by creating a new changeset. You can either use the console or the CreateKxChangeset API to create a changeset. To add a data to your database, create a changeset by providing the changeset type as PUT, database path, and S3 URI path.

To update data in a database, you need to create another changeset with the same database path you chose while adding the data. To delete data in a database, create a new changeset with changeset type as `DELETE`.

> **Note**
>
> You should only add data in the correct kdb file format that follows a valid kdb path structure. Other file formats and structures are not supported when accessed from a FinSpace Managed kdb cluster. You can learn more about valid kdb path structures here.

**To create a changeset from the console**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Databases** tab. The table under this tab displays a list of databases.
5. Choose a database name to view its details.
6. On the database details page, choose the **Changesets** tab.
7. Choose **Create changeset**.
8. On the **Create changeset** page, select one of the following types of changeset.
   - *PUT* – Adds or updates files in a database.
   - *DELETE* – Deletes files in a database. This option is not available when creating the changeset for the first time.
9. For **Database path**, specify a path within the database directory where you want to add data. If the data already exists at this path, it will be updated.
10. For **S3 URI** provide the source path of the file to add data.
11. Choose **Create changeset**. The database details page opens where you can see the status of the changeset in the changeset table.

    You can choose the changeset ID to view details of a changeset.

## Updating a kdb database

### To update the metadata of a kdb database

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Databases** tab.
5. From the list of databases, choose the one that you want to update. The database details page opens.
6. On the database details page, choose **Edit**.
7. Edit the database description.
8. Choose **Update database**.

## Viewing kdb database details

### To view and get details of a kdb database

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Databases** tab. The table under this tab displays a list of databases.
5. Choose a database name to view its details. The database details page opens where you can view details about the database. You can also add and view changesets and tags associated with this database.

## Deleting a kdb database

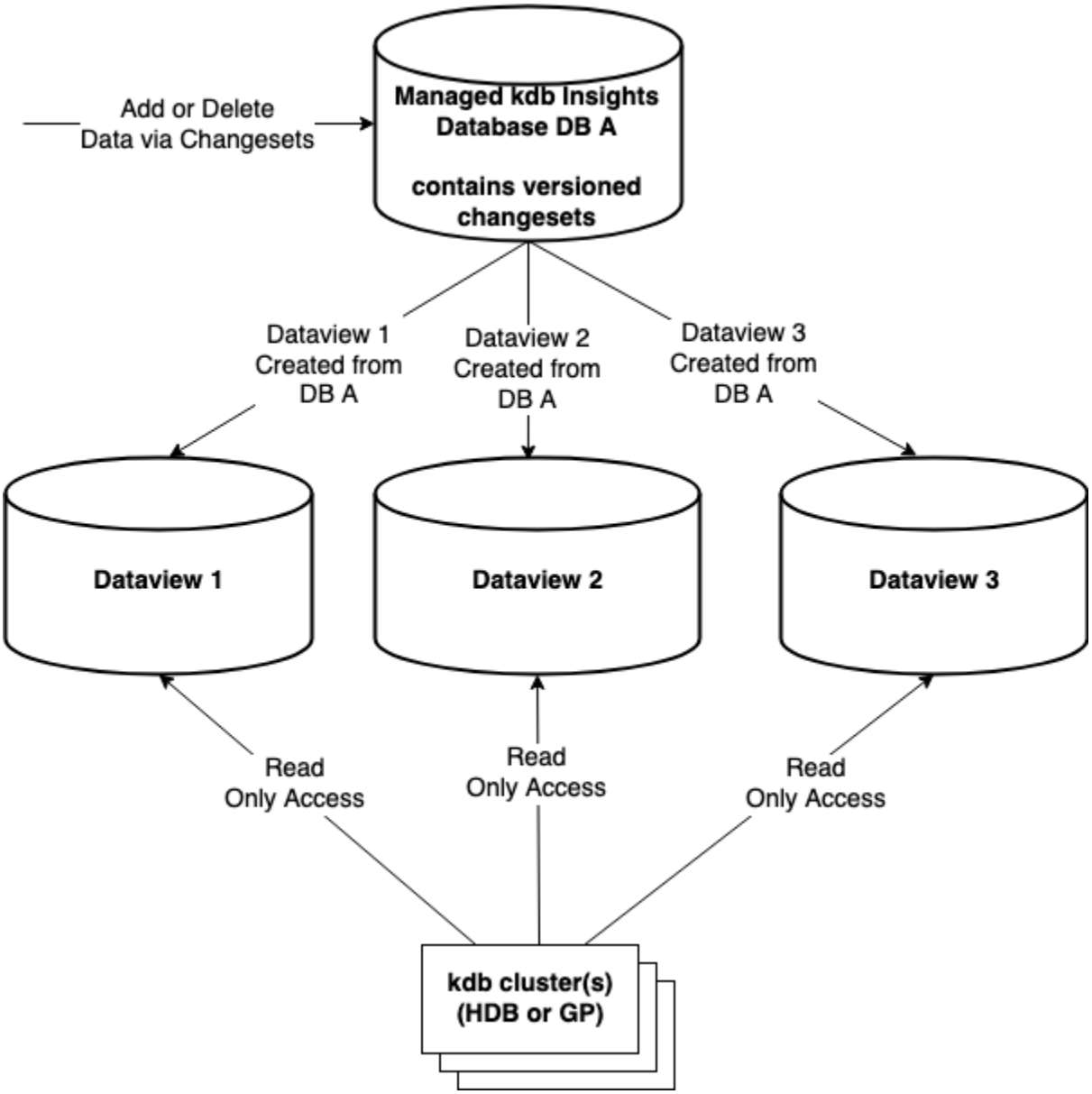
> **Note**
>
> This action is irreversible. Deleting a kdb database will delete all of its contents.

**To delete a kdb database**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Databases** tab.
5. From the list of databases, choose the one that you want to delete. The database details page opens.
6. On the database details page, choose **Delete**.
7. On the confirmation dialog box, enter *confirm*.
8. Choose **Delete**.

## Dataviews for querying data

Dataviews allow you to place portions of your Managed kdb Insights object store database onto disk for faster read-only access from your kdb clusters. To your kdb process, the dataview looks like a kdb segmented database, with data placed across one or more disk mounts (volumes) and the object store. This lets you place frequently-queried data on a fast-access disk for more performant access while keeping the rest of the data in the object store layer for less frequent access. With dataviews, the golden copy of your database's data still remains in the object store format. The data stored on disk for faster access is a copy.




Dataviews can be accessed from HDB and General purpose (GP) type clusters for read only access. The data within a dataview is accessible from the cluster as a kdb segmented database that is automatically configured when you associate the dataview with the cluster.

A segment is a mount point that can contain a portion of a database. Different segments could contain different data partitions, tables, or even columns. A kdb *par.txt* file that FinSpace automatically creates when you mount a database defines the segments.

The segments of this segmented database can reside on different kdb Insights disk volumes. A segment of your database can be any portion of it. For example, consider a database with contents as the following date-partitioned layout.

```
/sym
/2023.10.01/trades/price
/2023.10.01/trades/time
/2023.10.01/trades/quality
/2023.10.01/trades/price
/2023.10.02/trades/time
/2023.10.02/trades/quality
/2023.10.02/trades/price
/2023.10.03/trades/time
/2023.10.03/trades/quality
/2023.10.03/trades/price
/2023.10.04/trades/time
/2023.10.04/trades/quality
/2023.10.05/trades/price
/2023.10.05/trades/time
/2023.10.05/trades/quality
/2023.10.05/trades/price
/2023.10.01/trades/.d
/2023.10.02/trades/.d
/2023.10.03/trades/.d
/2023.10.04/trades/.d
/2023.10.05/trades/.d
```

In this example, `trades` is a table and `time`, `quantity`, and `price` are columns. You can store the most recent day of data on a high throughput volume, two days prior to that on 250 MB/s/TiB volume, with the rest accessible as a segment from the object store layer. The following table shows the data and segments.

| Database contents | Segments |
|---|---|
| /2023.10.05/trades/time | **Segment**: **Dataview Segment 1** |
| /2023.10.05/trades/quality | **Stored On**: Managed kdb Insights Volume 1 |
| /2023.10.05/trades/price | [High throughput – 1000 MB/s/TiB] |
| /2023.10.04/trades/time | **Segment**: **Dataview Segment 2** |
| /2023.10.04/trades/quality | **Stored On**: Managed kdb Insights Volume 2 |
| /2023.10.04/trades/price | [Medium Throughput – 250 MB/s/TiB] |

| Database contents | Segments |
| --- | --- |
| /2023.10.03/trades/time<br>/2023.10.03/trades/quality<br>/2023.10.03/trades/price | |
| /2023.10.02/trades/time<br>/2023.10.02/trades/quality<br>/2023.10.02/trades/price<br>/2023.10.01/trades/time<br>/2023.10.01/trades/quality<br>/2023.10.01/trades/price | **Segment**: **Dataview Default Segment**<br>**Stored On**: Object store |

This gives you control to place copies of portions of your database on the appropriate type of disk for access, if you require higher performance access than what is available with the default object store storage.

In addition, having the ability to explicitly place data on different volumes when creating a dataview, the contents directly under the root (/) path of the database, such as `/sym` in this example, are always copied to the cluster's local storage for fast access.

## Auto-updating vs static dataviews

When you create a dataview, you can specify from one of the following types of dataview.

- **Auto-updating** –An auto-update dataview contains the most recent version of the data in the database. Its contents are automatically updated as new data is added to the database.
- **Static** – For a static dataview, the data within the view is not updated automatically as new data is added to the database. When creating a static dataview, you specify a database version identifier that is the changeset ID. The dataview will contain contents of the database as of that changeset ID. To refresh the contents of a static dataview, you need to update it. If you do not provide a changeset ID when updating a dataview, system picks the latest one by default.

## Dataview versions

When you create a dataview, it is assigned an initial version. Each update, whether automatic or manual, creates a new version of a dataview. A dataview version becomes active when it is mountable. A dataview version is released when it is not attached to any clusters and when it's no longer the latest active version.

## Data placement

For each volume, you specify a list of paths for the data that you want to place on the volume. This can be done by using the db paths. Your paths can include the wildcard characters — asterisk (*) and question mark (?). Here are a few examples of how you can use db paths for segment configuration.

- To specify a particular partition – `/2020.01.02/*` or `/2020.01.02*`
- To specify all partitions for Jan 2020– `/2020.01.*` or `/2020.01*`
- To specify all partitions for 1st of each month in 2020 – `/2020.??.01` or `/2020.*.01`
- To specify all partitions – `/*` or `*`

## Data cardinality

You can create multiple dataviews for a single database. For example, you may wish to create one dataview based on an older version of the database for historical analysis, at the same time you may want an auto updating dataview for applications to query more recent data in your database. You can also use multiple dataviews with the same data in each, as a way to spread query load from a large number of clusters querying the data. You can create two different dataviews on the same changeset version.

## Consideration

- Dataviews are only available for clusters running on a scaling group. They are not supported on dedicated clusters.
- The paths placed on different volumes cannot overlap. For example, you could not place a path of `/2023.10.31/*` on one volume of a dataview and `/2023.10*` on another volume of the same dataview because the paths overlap. This constraint is because each volume is a different segment in the `par.txt` file on the database and contents of a segment can’t overlap.

## Managing kdb dataviews

The following sections provide a detailed overview of the operations that you can perform by using a Managed kdb dataview.

### Creating a kdb dataview

#### To create a kdb dataview

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Databases** tab.
5. On database details page, choose the **Dataviews** tab.
6. Choose **Create dataview**.
7. On the **Create dataview** page, enter a unique name for the dataview.
8. (Optional) Enter a description for your dataview.
9. Choose the availability zone that you want to associate with the dataview. Currently, you can only choose single availability zone.
10. Choose a how you want to update data in the dataview from one of the following options.

    - **Auto-update** – Adds the most recent version of the data in a database. The dataview is automatically updated as new data is added to the database.
    - **Static** – Add data based on the changeset ID that you specify. The dataview is not automatically updated as new data is added to the database. To refresh the contents of a static dataview, you need to update it and specify a new changeset ID. When you choose this, the **Read Write** option enables.

    a. If you choose **Static**, specify the **Changeset ID** to indicate which version of data you want.

    b. If you choose **Static**, you get the option to make dataviews writable. Select **True** if you want to make the dataview writable to perform database maintenance operations. By default, this value is set to **False**. For more information, read this section.

11. (Optional) For **segment configuration**, specify the database path of the data that you want to place on each selected volume. You can also enable **On demand caching** on the selected database path when a particular file or a column of a database is accessed.

    > **Note**
    >
    > - Each segment must have a unique database path for each volume.
    > - Every data view has a default segment associated with it. The default segment is S3/ object store segment. All database paths not explicitly specified to be on a volume are accessible from the cluster through this default segment.
    > - The **Segment configuration** is required if **Read Write** is **True**. You can only add one segment for a writeable dataview.
    > - The **Database path** is disabled and defaults to ***** when **Read Write** is **True** as you cannot have partial writeable dataviews on cache.

12. Choose **Create dataview**. The database details page opens and the table under **Dataviews** lists the newly created database along with its status.

    You can choose the dataview name from the list to view its details.

### Viewing kdb dataview details

#### To view and get details of a kdb dataview

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Databases** tab.
5. From the list of databases, choose a database name. The database details page opens.
6. On the database details page, choose the **Dataviews** tab that shows a list of dataviews along with its status, availability zones where they were created, and their creation time.
7. From the list of dataviews, choose a name to view its details. The dataviews details page opens where you can view the following details.

    - **Dataview details** section – Displays the metadata of the dataview that you created.

- **Configuration** tab – Displays the details about the dataview update mode and ID, and the availability zones ID.
- **Active versions** tab – Displays a list of active versions of the dataview. Each update of the dataview creates a new version, including changeset details and the cache configurations. Each version triggers a workflow to cache database based on the cache configuration. A dataview version becomes active once the workflow finishes.

  The dataview version is deactivated under the following conditions

  - It's not the latest active version.
  - No cluster is currently mounting this version.

  You can choose the **Version ID** to see details of each active version.
- **Clusters** tab – Displays a list of clusters that mounts the dataview.

### Updating a kdb dataview

#### To update a kdb dataview

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Databases** tab.
5. From the list of databases, choose a database name. The database details page opens.
6. On the database details page, choose the **Dataviews** tab.
7. From the list of dataviews, choose a name and then choose **Edit**.
8. On the edit page, you can update the description for the dataview. If the dataview is **Static**, you can also update the **Changeset ID**.
9. Choose **Save changes**.

### Deleting a kdb dataview

Before deleting a dataview, make sure that it is not in use by any cluster. You can check this from the **Clusters** tab in the dataview details page.

> **Note**
>
> This action is irreversible. Deleting a kdb dataview will delete all of its metadata.

**To delete a kdb dataview**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Databases** tab.
5. From the list of databases, choose the one whose dataview you want to delete. The database details page opens.
6. On the database details page, choose the **Dataviews** tab.
7. From the list of dataviews, choose a name and then choose **Delete**.
8. On the confirmation dialog box, enter **confirm** to provide a written consent to delete the resource permanently.
9. Choose **Delete**.

## Database maintenance

Amazon FinSpace Managed kdb allows you to perform schema changes to your database like adding a new column, updating a column type, and renaming columns, etc. You can perform the database maintenance operations by creating a general purpose cluster with a writable dataview. A writable dataview allows you to make updates to your kdb database locally on a cluster. To avoid caching the whole kdb database on a cluster, you can enable on-demand caching for your dataview segments. The dataview will only load the filesystem metadata of your database files for the segments with on-demand caching and loads the actual file content as they are accessed by a database maintenance operation.

You can implement a database maintenance script and run it as an initialization script. An initialization script can run for multiple hours without being interrupted, which is required for long-running database maintenance tasks. When database maintenance script is running, monitor the cluster logs for progress and any errors. After the database maintenance script completes,

connect to the cluster to verify the updated kdb database and commit changes by using the `commit_kx_database` q API. The API creates a changeset and returns the changeset id, which you can use to monitor the changeset status through either the FinSpace API or console. You can also automate verification and commit steps in your database maintenance script itself. For more information, see the following sections.

**Topics**



## Setting up for database maintenance

To perform the database maintenance operations, you need a writeable shallow copy of a database . A writable shallow copy of a kdb database only loads the metadata of your database files to make them visible on the file system and loads the actual file content as they are accessed. To optimise time and memory utilization, it is recommended not to load file content initially, as not all files may be necessary for a database maintenance operation. For instance, in the case of renaming a table, no files are read or updated directly.

To create a writeable shallow copy of database, you can create dataviews with read write property set as true and enable on-demand caching in the configuration. A dataview performs minimal loading of files on the file system as needed by a database maintenance operation when on-demand caching is enabled. Reading an existing database file for the first time is slower as compared to accessing the files that have been previously read or newly written. This is because files are loaded onto the file system as they are accessed in case of on-demand caching.

### Creating writeable dataviews by using console

Before you proceed, complete the following prerequisites:

- Create a kdb environment. For more information, see Creating a kdb environment.
- Create a kdb database. For more information, see Creating a kdb database.
- Create a new changeset. For more information, see Creating a new changeset.
- Create a kdb volume. Make sure this volume is not used by any other resource. For more information, see Creating a Managed kdb volume.

**To create a writeable dataview**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Databases** tab.
5. From the list of databases, choose a database name.
6. On database details page, choose the **Dataviews** tab.
7. Choose **Create dataview**.
8. On the **Create dataview** page, enter a unique name for the dataview.
9. (Optional) Enter a description for your dataview.
10. Choose the availability zone that you want to associate with the dataview. Currently, you can only choose single availability zone.
11. Under **Changeset update settings**, do the following.

    a. Choose **Static** mode of update.

    > **Note**
    >
    > The **Read Write** option is only available for **Static** update mode as you cannot perform automatic updates on a writeable dataview.

    b. Select the **Changeset ID** for the changeset you created to indicate which version of data you want.

    c. Choose **Read Write** as **True** to make this dataview as writeable. You cannot change this later.

12. Add **Segment configuration**.

> **Note**
>
> - The **Segment configuration** is required if **Read Write** is **True**.
> - You can only add one segment for a writeable dataview.

- The **Database path** is disabled and defaults to ***** when **Read Write** is **True** as you cannot have partial writeable dataviews on cache.

a. Choose a volume for caching. Use an exclusive volume for writable dataviews, it should not be in use by any other dataviews.

b. For **On demand caching**, choose **True** to enable on demand caching on the selected database path when a particular file or a column of a database is accessed. When you enable on demand caching, files will only be copied to the dataview when they are accessed by code for reading or writing. When you disable on demand caching, everything is cached. The default value is **False**.

13. Choose **Create dataview**. The database details page opens and the table under **Dataviews** lists the newly created database along with its status.

### Creating writeable dataviews by using FinSpace API operations

Before you proceed, complete the following prerequisites:

- Create a kdb environment by using the CreateKxEnvironment API operation.
- Create a kdb database by using the CreateKxDatabase API operation.
- Create a new changeset by using the CreateKxChangeset API operation.
- Create a kdb volume by using the CreateKxVolume API operation. Make sure this volume is unique for this dataview and is not used by any other resource.

To create a dataview with writable shallow copy of a database, create a dataview with the volume that has writable segments by using the CreateKxDataview API operation. You can make dataview as writeable by setting the `readWrite` parameter as true. You can only use this parameter for a static update mode. The `onDemand` parameter allows you to enable or disable on-demand caching on the selected dbPaths.

Sample `CreateKxDataview` API request

```
{
    "autoUpdate": false,
    "availabilityZoneId": "use1-az1",
    "clientToken": "65117136-4421-4371-0f1a-ce012823126",
```

```
    "changesetId": "latest_changesetId",
    "readWrite": true,
    "segmentConfigurations": [{
            "volumeName": "test_vol",
            "dbPaths": [
                "/*"
            ],
            "onDemand": true
        }

    ],
    "azMode": "SINGLE",
    "dataviewName": "test_dv"
}
```

Following are some of the considerations for the above request.

- The `autoUpdate` must be `false` for if `readWrite` is `true` on the dataviews.
- You need exclusive volume for creating a writable dataview. The volume mentioned in the `segmentConfiguration` should not be used by any other dataview.
- The `dbPath` must be set as "/*" for writable dataview.
- Only a single `segmentConfiguration` is allowed when `readWrite` is true. The `dbPaths` on the segment should be set as "*" .
- A dataview with `readWrite` set as `true` is not allowed to be updated.
- You cannot update the `readWrite` property later.
- A dataview can only have a single segment if `onDemand` is `true` on a segment.

## Performing database maintenance

After you create a writeable dataview, you create a scaling group general purpose cluster to run a long-running database maintenance script. For this, you use the cluster `initializationScript` attribute. The database maintenance script could run for multiple hours without being terminated. When database maintenance script is running, monitor the cluster logs for progress and any errors from the database maintenance script. After the database maintenance script completes, connect to the cluster to verify the updated kdb database and commit changes to the underlying kdb database by using the `commit_kx_database` q API. You can also automate these steps in your database maintenance script itself.

**Steps to perform database maintenance using a scaling group general purpose cluster**

1. Create a general purpose cluster in the scaling group with the previously created data view and provide database maintenance script using `initializationScript` in the CreateKxCluster API operation. After you create the cluster, wait till the status changes to `Running`. During this time, you can monitor the logs from the cluster for progress and any errors from the database maintenance script.
2. Call the GetKxConnectionString API to get a `signedConnectionString` for the cluster.
3. Connect to the cluster and verify the kdb database state by running q commands.
4. Call the `commit_kx_database` q API with the database name to apply the changes to the source kdb database.
5. Call the GetKxChangset API operation to check the status of the commit database changeset. After the kdb database is successfully updated, you can load the updated kdb database on an existing HDB cluster by calling the UpdateKxClusterDatabases API operation or on a new HDB cluster by calling the CreateKxCluster API operation.

**Steps to perform database maintenance using `dbmaint.q`**

This is section shows how you can perform database maintenance on a partitioned database by using a `dbmaint.q` script. The following example explains how you can load the `dbmaint.q` script on a general purpose cluster that runs on a scaling group, add a new column to a table, and finally commit the database to create a changeset.

1. Load the `dbmaint.q` script by running the following command. This script contains utility functions for maintenance of partitioned database tables in kdb+.

```
q) \l /opt/kx/app/code/dbmaint/dbmaint.q
```

2. Load a database.

```
q) \l /opt/kx/app/db/welcomedb
```

3. Inspect the table schema in your database.

```
q) meta example
c     | t f a
------| -----
date  | d
```

```
sym   | s   p
time  | p
number| j
```

4. Change to the database parent directory.

```
q) \cd /opt/kx/app/db
```

5. Add a new column using the `addcol` function from the `dbmaint.q` script.

```
addcol[`:welcomedb;`example;`price;0h];
```

6. Inspect the updated table schema with the newly added column.

```
q)con "meta example"
c     | t f a
------| -----
date  | d
sym   | s   p
time  | p
number| j
price | h
```

7. Commit the database changes by calling the `.aws.commit_kx_changeset q` API. The API creates a changeset and returns the id, which you can use to monitor the changeset status through the FinSpace API or console.

```
q) .aws.commit_kx_database["welcomedb"]
id    | "UscXQcZ2htijCQlr1xNaIA"
status| "PENDING"
```

**Note**

The recommended way to perform a long-running database maintenance is to implement a database maintenance script and execute it as cluster initialization script. An initialization script can run for multiple hours without being interrupted which is required for long-running database maintenance tasks. When database maintenance script is running, monitor the cluster logs for progress and any errors. After the database maintenance script completes, connect to the cluster to verify the updated kdb database and commit changes

to the underlying kdb database by using the commit_kx_database q API. You can also automate verification and commit steps in your database maintenance script itself.

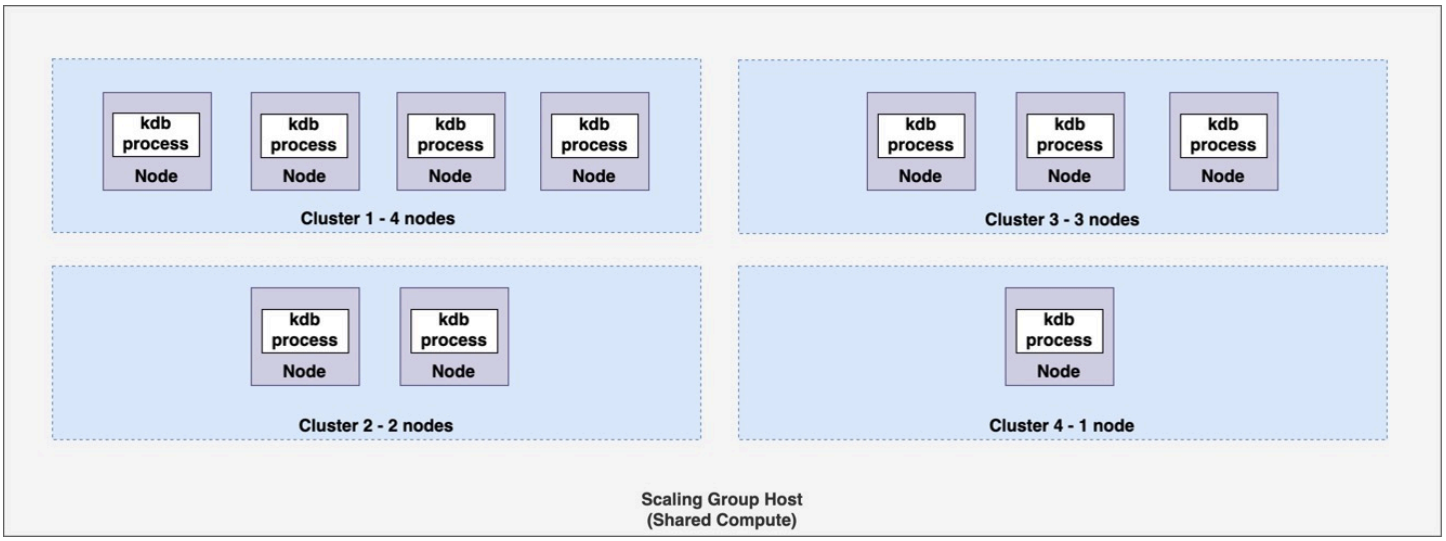
# Managed kdb scaling groups

Many kdb customers today use a deployment architecture consisting of multiple kdb processes running on a single compute host. When workloads are such that the resource of demands of the different processes don't conflict, this approach can maximize use of computing resources (CPU, RAM, I/O) to achieve more efficient use of computing resources. Scaling groups allows you to take this same approach with Managed kdb Insights.

Scaling group terminology

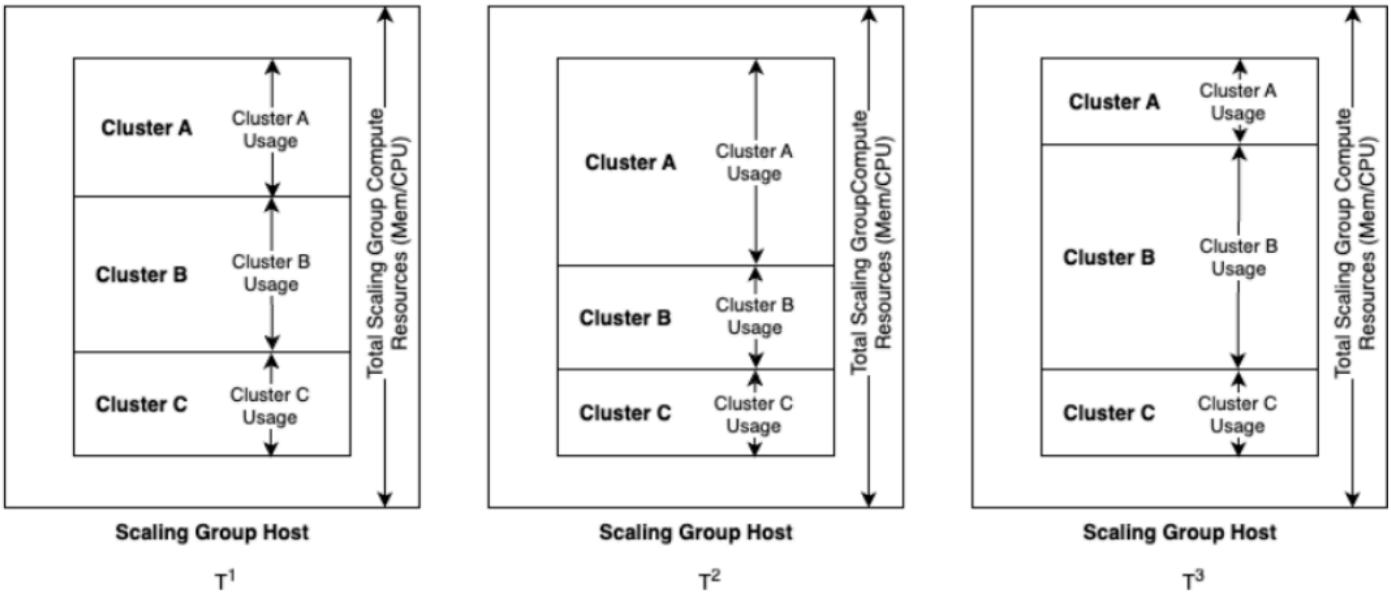
- Scaling group – Shared compute you can run your kdb workloads (clusters) on.
- Scaling group host – A single unit of compute in a scaling group. Scaling groups currently can only have a single host.
- Cluster – A set of one or more identically configured kdb process (nodes).
- Cluster node – A single kdb process, running within a cluster.

With scaling groups, you can run multiple kdb workloads or clusters on shared compute (a host) that you provision. This allows you to maximize utilization of compute in your FinSpace Managed kdb Insights environment. You can run multiple clusters on a single scaling group host. Each cluster can have one or more nodes, each with a kdb process.



The previous diagram is an example of four clusters running on a scaling group host. Cluster 1 has four nodes, Cluster 2 has two nodes, cluster 3 has three nodes and cluster 4 has one node. As memory requirements for an individual cluster vary throughout the day, each may consume different amounts of memory. By placing workloads that have memory needs that peak at different times throughout the day, you can place more workloads or clusters in a fixed set of compute than it is possible if you used FinSpace dedicated cluster option.

For example, you may have multiple HDB workloads where memory requirement of any individual HDB will vary at different times of the day, but in total they will all remain within a certain known memory footprint. You can place all of these workloads onto a scaling group to share resources like CPU and memory as shown in the following diagram.




## High level workflow for running clusters on a scaling group

Before running a cluster on a scaling group, you need to create the scaling group itself. Once you create the scaling group, you can launch one or more clusters on it. You can display clusters running on a scaling group by using the `ListKxClusters` API or from the **Clusters** tab in Amazon FinSpace console. When you delete a cluster running in a scaling group, the host and any other running clusters on the scaling group remain available. If there are no clusters running on a scaling group, you may delete it.

## Resource management with scaling groups

When launching a cluster to run on a scaling group,the total available amount of memory on the scaling group host is limited. The following table describes the limits of each host.

| Compute type | vCPUs | Memory available for kdb (GiB) |
|---|---|---|
| kx.sg.large | 2 | 16 |
| kx.sg.xlarge | 4 | 32 |
| kx.sg.2xlarge | 8 | 64 |
| kx.sg.4xlarge | 16 | 108 |
| kx.sg.8xlarge | 32 | 216 |
| kx.sg.16xlarge | 64 | 432 |
| kx.sg.32xlarge | 128 | 864 |
| kx.sg1.16xlarge | 64 | 1949 |
| kx.sg1.24xlarge | 96 | 2948 |

When launching a kdb cluster to run on a scaling group, you specify the minimum memory required for each kdb process in the cluster (node) as well as expected amount of memory. If there is insufficient memory on the scaling group host to meet this required value, the cluster will not start. You can also specify an expected value for the amount of memory the cluster will require. The scheduler will use this to avoid launching the cluster if the memory value is not sufficient. For clusters with more than one node or kdb processes, the amount of memory used is the sum of the kdb process memory that each node consumes.

## Considerations

- Currently, a scaling group consists of a single scaling group host and clusters can only run on one scaling group at a time. If you need to run more clusters in your environment than can fit on a

single scaling group host, you may run multiple and put different clusters from your set on to different scaling groups.

- You cannot delete a scaling group until you delete all the clusters running on it.
- Savedown storage does not work with General purpose (GP) and RDB clusters running on scaling groups. Instead, you should use volumes for the temporary storage of your savedown data.
- HDB and GP clusters, when they are run as a part of a scaling group, don't support high performance HDB disk cache. You may instead use dataviews if you need to place portions of your database on high performance disk.

# Managing kdb scaling groups

The following sections provide a detailed overview of the operations that you can perform by using Managed kdb scaling groups.

**Topics**



## Creating a Managed kdb scaling group

**To create a scaling group**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Kdb scaling groups** tab.
5. Choose **Create kdb scaling group**.
6. On the **Create kdb scaling group** page, enter a unique name for the scaling group details.
7. Choose a **Host Type** based on the available throughput and size.
8. Choose the availability zone that you want to associate with the scaling group. Currently, you can choose only single availability zone.

9. (Optional) Add a new tag to assign it to your scaling group. For more information, see AWS tags.

> **Note**
>
> You can only add up to 50 tags to your user.

10. Choose **Create kdb scaling group**. The scaling group creation process starts and the kdb environment details page opens where you can see the status of creation under the **Kdb scaling groups** tab.

## Viewing a Managing kdb scaling group

### To view and get details of a Managing kdb scaling group

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Kdb scaling groups** tab.
5. From the list of scaling groups, choose a name to view its details. The Kdb scaling group details page opens where you can view the following details.

   - **Scaling group details** section – Displays the metadata of the scaling group that you view.
   - **Configuration** tab – Displays the availability zone for the scaling group.
   - **Monitoring** tab – Displays the dashboard of scaling group metrics. You can view activity logs for your scaling group here.
   - **Clusters** tab – Displays a list of clusters running on this scaling group. For information on how to create clusters, see Creating a Managed kdb Insights cluster.
   - **Tags** tab – Displays a list of key-value pairs that are associated with the scaling group. If you did not provide tags during volume creation, choose **Manage tags** to add new tags.

### Deleting a Managing kdb scaling group

> **Note**
>
> This action is irreversible. Deleting a scaling group will delete all of its data.

#### To delete a Managing kdb scaling group

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Kdb scaling groups** tab.
5. From the list of scaling groups, choose the one that you want to delete and choose **Delete**. Alternatively, you can choose the scaling group name and open the details page to delete it.
6. On the confirmation dialog box, enter **confirm** to provide a written consent to delete the resource permanently.
7. Choose **Delete**.

# Managed kdb volumes

Volumes are managed storage that resides in your Managed kdb Insights environment and can be associated with clusters for storage of data such as TickerPlant (TP) logs, Real-time Database (RDB) savedown files, and temporary storage on General purpose (GP) type clusters. Volumes can also be used by dataview to store copies of your database of disk for fast access when reading data from a database. You can also use volumes to share data between RDB and Intra-day DB (IDB).

## Volumes for temporary data storage

You can use a Managed kdb volume data storage for your cluster. When creating a TP cluster, you must specify a volume that will hold the TP logs. For an RDB or GP cluster running on a scaling group, you can specify a volume to hold savedown or temporary files.

Multiple clusters can share a single volume for simplicity as shown in Figure 1, or you can configure multiple volumes and associate them with specific clusters for workload isolation as shown in Figure 2.

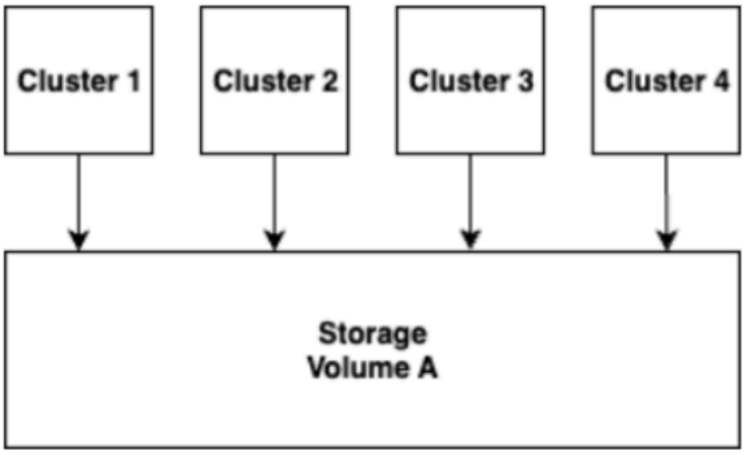



*Figure 1*

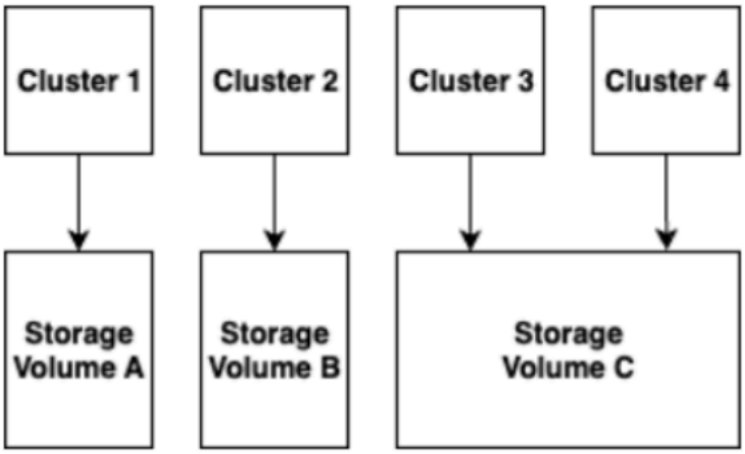



*Figure 2*

## Volumes with dataviews

You can use Managed kdb volumes when you create dataviews. Dataviews store a copy of the data in your database on one or more volumes for fast access. When creating a dataview you can specify one or more volumes to store a portion of your database for faster data access compared to querying the data from the default object store format of the data in the database. For more information about using volumes as part of a dataview, see Creating a kdb dataview.

## Volume types

Volumes are available in different price or performance characteristics based on your need. Currently FinSpace offers the following three types of volume with three throughput characteristics.

- 1000 MB/s/TiB –
- 250 MB/s/TiB
- 12 MB/s/TiB

## Considerations

- When you delete a cluster, the data remains on the volume. If you don't want this delete data before deleting the cluster.
- You can access data mounted on a volume from within a cluster from the path `/opt/kx/app/shared/$VOLUME_NAME/$CLUSTER_NAME`.

## Managing kdb volumes

The following sections provide a detailed overview of the operations that you can perform by using Managed kdb volumes.

**Topics**



### Creating a Managed kdb volume

**To create a Managed kdb volume**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.

3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Volumes** tab.
5. Choose **Create volume**.
6. On the **Create volume** page, enter the volume details and choose the **Volume type**. Currently, FinSpace only supports **NAS_1** (network attached storage) volume type.
7. Choose the throughput from one of the following types.
   - **SSD_1000**
   - **SSD_250**
   - **HDD_12**
8. Enter the size for the network attached storage configuration. For storage type **SSD_1000** and **SSD_250** you can select the minimum size as 1200 GB or increments of 2400 GB. For storage type **HDD_12** you can select the minimum size as 6000 GB or increments of 6000 GB.
9. Choose the availability zone that you want to associate with the volume.
10. (Optional) Add a new tag to assign it to your Managed kdb volume. For more information, see AWS tags.

    > **Note**
    >
    > You can only add up to 50 tags to your user.

11. Choose **Create volume**. The volume creation process starts and kdb environment details page opens where you can see the status of volume creation under the **Volumes** tab.

## Viewing a Managed kdb volume

### To view and get details of a Managed kdb volume

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Volumes** tab. The table under this tab displays a list of volumes created in the environment.

5. Choose a volume name to view its details. The volume details page opens where you can view the following details.

   - **Volume details** section – Displays the metadata of the volume that you created.
   - **Configuration** tab – Displays the details about the network attached storage and availability zones.
   - **Monitoring** tab – Displays the dashboard of volume metrics. You can view activity logs for your volume here.
   - **Clusters** tab – Displays a list of clusters attached to this volume. For information on how to create clusters, see Creating a Managed kdb Insights cluster.
   - **Tags** tab – Displays a list of key-value pairs associated with the volume. If you did not provide tags during volume creation, choose **Manage tags** to add new tags.

## Updating a Managed kdb volume

You can only edit the description and size of a volume. When you update a volume, you can only increase the volume size but cannot reduce it. During the update process, the filesystem might be unavailable for a few minutes. You can retry any operations after the update is complete.

**To update a Managed kdb volume**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the list of environments, choose a kdb environment.
4. On the environment details page, choose the **Volumes** tab.
5. From the list of clusters, choose the one that you want to edit. The volume details page opens.
6. Choose **Edit** and update the required details.
7. Choose **Save changes**.

### Deleting a Managed kdb volume

> **Note**
>
> This action is irreversible. You cannot delete a volume if it's attached to any cluster or dataview.

#### To delete a Managed kdb volume

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Volumes** tab.
5. From the list of volumes, choose the volume that you want to delete and choose **Delete**. Alternatively, you can choose the volume name and open the volume details page to delete it.
6. On the confirmation dialog box, enter **confirm** to provide a written consent to delete the resource permanently.
7. Choose **Delete**.

## Managed kdb Insights clusters

A FinSpace Managed kdb Insights cluster is a set of compute resources that run kdb processes in a FinSpace Managed kdb environment. By using FinSpace Managed kdb clusters, you can easily set up your own private managed data processing and analytics hub for capital markets data. This provides access to real-time and historical data along with high-performance analytics.

#### Topics

# Running a clusters on scaling groups vs as a dedicated cluster

The original Amazon FinSpace Managed kdb cluster launch configuration is now referred to as a dedicated cluster. In a dedicated cluster, each node or kdb process in the cluster runs on its own dedicated compute host.

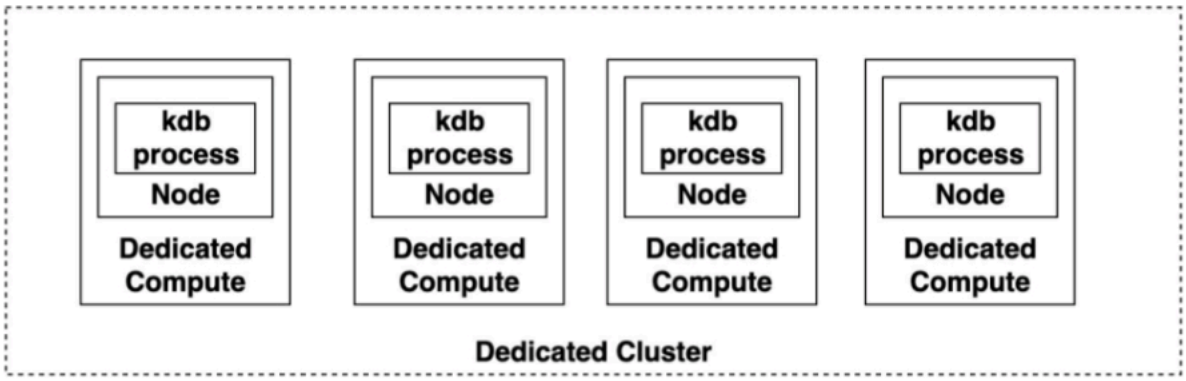
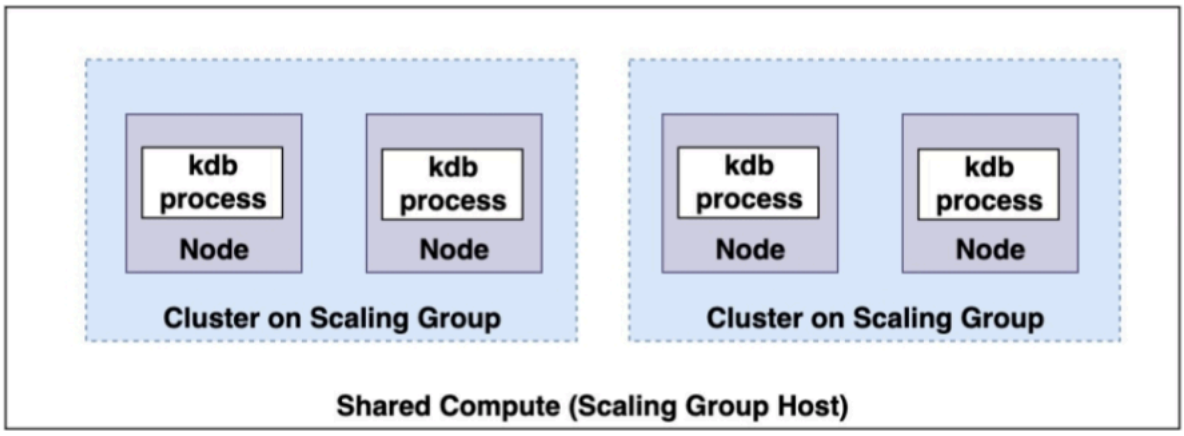



*Figure 1*

This configuration provides strong workload isolation between clusters and nodes in a single cluster at the expense of needing a fixed amount of compute per node. In contrast, with a cluster on scaling group a single set of compute is shared by multiple workloads (clusters) running on shared compute, allowing you to share a fixed amount of compute.




### Considerations

- Currently, a kdb scaling group is limited to only one host residing in one Availability Zone.
- The HDB clusters running on kdb scaling groups must use dataviews instead of cluster-specific disk cache to store database data for high-performance read access.

- RDB and General Purpose clusters running on scaling groups must use a kdb volume for their savedown storage configuration.

## Cluster types

Amazon FinSpace supports a variety of kdb clusters that you can use for different uses cases such as to implement a standard kdb tick architecture.

### General purpose

You can use a *general purpose* cluster if your kdb application doesn't require any specific features that are available on more specialized clusters—like the multi-node, Multi-AZ read-only query of an HDB cluster or the multi-node, Multi-AZ gateways.

With a general purpose cluster, you can mount a kdb Insights database for read-only access, as well as storage (savedown storage) for writing. This ability to read a database and write contents from a single cluster makes general purpose clusters suitable for various maintenance tasks. For example, you can use a general purpose cluster for tasks that require the ability to read and write data, and for creating derived datasets from an HDB cluster, in support of use cases such as one-time analysis by quantitative analysts (quants).

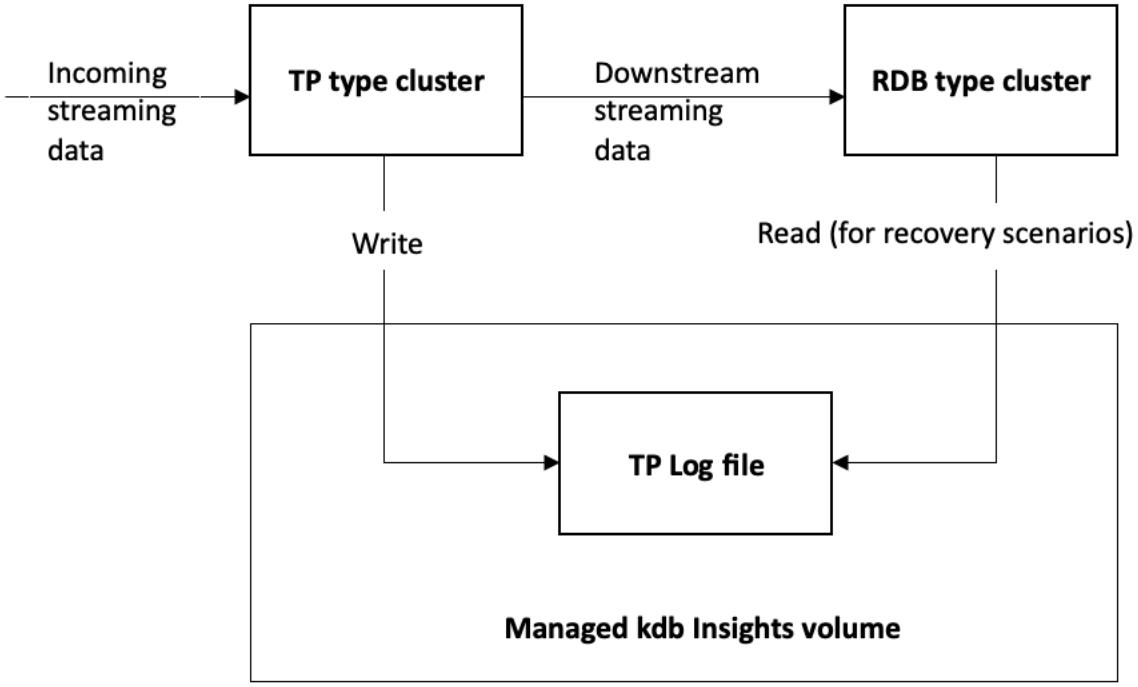
#### Features of a general purpose cluster

The following are the features of a general purpose cluster.

- The node count for this cluster type is fixed at 1.
- It only supports Single-AZ mode.
- It can mount a kdb Insights database for read-only access to data.
- You can configure savedown storage at the time of creating the cluster. You can use this space for writing savedown files before loading into a FinSpace database, or as a writeable space of other temporary files. For dedicated clusters, the savedown storage becomes unavailable when the cluster node is deleted.
- For clusters running on a scaling group, the savedown storage location will use a shared volume. This volume exists even after you delete the cluster and can be used by other clusters. You can remove the data on the volume before deleting the cluster or it remains available for use by other clusters.
- It can update databases and cache with the UpdateKxClusterDatabase operation.

## Tickerplant

A tickerplant (TP) acts as a message bus that subscribes to data or gets data pushed to it by Feed Handlers and then publishes it to one or more consumers, typically a realtime database (RDB). It persists a copy of each message received to a durable log of messages that are called the TP Log, so that downstream subscribers can request a replay of messages if needed. The following diagram explains that you can configure a TP cluster to save logs to a volume in Managed kdb Insights, from where you can replay the logs from an RDB type cluster.




### Features of a tickerplant cluster

Following are the features of a tickerplant type cluster:

- It supports only single-node that is only one kdb process.
- It shares storage with RDB clusters.
- It does not support the Multi-AZ mode. If you need Multi-AZ redundancy, run two TP type clusters in parallel.

## Gateway

In the vast majority of kdb+ systems, data is stored across several processes, which results in the need to access data across these processes. You do this by using a gateway to act as a single interface point that separates the end user from the configuration of underlying databases or services. With a gateway, you don't need to know where data is stored, and you don't need to make multiple requests to retrieve it.

To support running your custom gateway logic, Managed kdb Insights provides a *gateway* cluster type. You can deploy your own routing logic using the initialization scripts and custom code. You can configure gateways to a multi-node, Multi-AZ deployment for resiliency.

**Features of a gateway cluster**

The following are the features of a gateway type cluster:

- It provides support to run gateways with your custom allocation hosted inside of a Managed kdb environment.
- It provides support for hosting code with custom allocation logic for allocating load across different kdb clusters or nodes.
- It integrates with the discovery service to understand available clusters, monitor their health status, and provide an endpoint for the cluster.
- It provides a network path from your custom code running on the gateway to the cluster supporting IPC connections.

## Real-time database (RDB)

You can use a *real-time database* cluster to capture all the data from another kdb process, such as a ticker plant, and store it in memory for query or real-time processing. Because the data volume can eventually exceed the amount of available memory, kdb customers typically move the data from the RDB to a historical database (HDB) using a process called *savedown*. This process typically occurs at the end of a business day.

You can create, list, and delete RDB clusters with single or multiple nodes through both console and FinSpace API operations.

### Savedown storage

RDB clusters require local space for temporary storage of data during the savedown process. This temporary storage is used to hold data for the period between when a cluster has flushed it from memory and when it is successfully loaded into a kdb Insights database. To support this, RDB clusters have writeable disk that is used as storage space for savedown data. You can use the data saved down to the FinSpace database from and RDB by creating an HDB cluster that points to the database.

### Considerations

The following are some considerations related to savedown storage:

- You can configure savedown storage at the time of creating the cluster. You can use this space to write savedown files before loading into a FinSpace database, or as a writeable space for other temporary files.
- For dedicated clusters, the savedown storage becomes unavailable when you delete a cluster node.
- For clusters running on a scaling group, the savedown storage location will use a shared volume. This volume exists even after you delete the cluster and can be used by other clusters. You can remove the data on the volume before deleting the cluster or it remains available for use by other clusters.

## Historical database (HDB)

A historical database holds data from a day before the current day. Each day, new records are added to the HDB at the end of day. To access data in Managed kdb databases from an HDB cluster, you must attach the databases you want to access as an option when launching the cluster. You can do this at the time of creating a cluster through the console, or by using the create cluster API operation in the *Amazon FinSpace Management API Reference*. The HDB cluster can access this data in a read-only mode.

### Cache configuration

When you attach a database to a cluster for access, by default, the read operations are performed directly against the object store that the database data is stored in. Alternatively, you can also define a file cache, in which you can load data for faster performance. You do this by specifying cache configuration when you associate the database with the cluster. You can specify a certain amount of cache, and then separately specify the contents of the database that you want to cache.

FinSpace supports the following cache types:

- **CACHE_1000** – This type allows a throughput of 1000 MB/s per unit storage (TiB).
- **CACHE_250** – This type allows a throughput of 250 MB/s per unit storage (TiB).
- **CACHE_12** – This type allows a throughput of 12 MB/s per unit storage (TiB).

### Considerations

The following are some considerations related to storage and billing:

- Caching is only available on dedicated clusters. For clusters running on a scaling group, use dataviews.
- You can only configure initial cache size at the time of cluster creation. To run a cluster with a different sized cache, you need to terminate the cluster and launch a new one with smaller database cache size.
- Billing for cache storage starts when storage is available for use by the cluster and stops when the cluster is terminated.

### Auto scaling

With the HDB auto scaling feature, you can take away some nodes to save costs when the usage is low, and add more nodes to improve availability and performance when the usage is high. For auto scaling HDB clusters, you specify the CPU utilization targets for your scaling policy. You can auto scale an HDB cluster at the time of cluster creation in two ways. You can use the console or use the createKxCluster API operation, where you provide minimum and maximum node count, the metric policy, and a target utilization percentage. As a result, FinSpace scales in or scales out the clusters based on service utilization that's determined by CPU consumed by the kdb+ node.

> **Note**
>
> Auto scaling is only available for dedicated clusters and is not supported for clusters running on scaling groups.

### Summary of capabilities by cluster type

| Capability | General purpose | Gateway | RDB | TP | HDB |
|---|---|---|---|---|---|
| Attaches a Managed kdb Insights database for read-only access | Yes | No | No | No | Yes |
| Attaches writable local (savedown) storage to a node | Yes | No | Yes | No | No |
| Number of nodes supported | Single | Multi | Multi | Single | Multi |
| Supports AZ configurations (for dedicated clusters) | Single | Single or Multi | Single or Multi | Single | Single or Multi |

## Managing kdb clusters

The following sections provide a detailed overview of the operations that you can perform by using Managed kdb clusters.

### Topics

- Creating a Managed kdb Insights cluster
- Viewing kdb cluster detail
- Updating code configurations on a running cluster
- Updating a kdb cluster database
- Deleting a kdb cluster

## Activating your Managed kdb Insights license

To run Managed kdb Insights clusters, you must first have an existing kdb Insights license from KX. That kdb Insights license needs to be activated for your Managed kdb Insights environment(s). You're responsible for working directly with KX (KX Systems, Inc., a subsidiary of FD Technologies plc) to obtain this.

To activate an existing kdb Insights license for your Managed kdb Insights environment(s), do the following:

- Contact your KX account manager or KX sales representative and provide them with the AWS account number for all the accounts where you want to use your Managed kdb Insights environment(s).
- Once arranged with KX, the kdb Insights license will be automatically applied to your Managed kdb Insights environment(s).

> **Note**
>
> If you do not have an existing kdb Insights license, you can request a 30-day trial license from KX here. KX will then activate a 30-day trial license for you.

- You will receive an activation email and your 30 day trial license will be automatically applied to your Managed kdb Insights environment.

> **Note**
>
> If KX has already enabled a kdb license for use with Managed kdb Insights in your AWS account and the license has not expired, you can start using clusters in your environment as soon as it is created. You do not need to request a new license.

## Managed kdb Insights cluster software bundles

When you launch a cluster, you can choose the software versions that will run on your cluster. This allows you to test and use application versions that fit your compatibility requirements.

You can specify the release version using the `Release Label`. Release labels are in the form `x.x`.

The following table lists the software versions that each release label includes. Currently it only includes the kdb Insights core.

| Managed kdb Insights release label | Kdb Insights core version | |
|---|---|---|
| 1.0 | 4.0.3 | |

## Maintaining a Managed kdb Insights cluster

Maintaining a kdb cluster involves updates to the cluster's underlying operating system or to the container hosting the Managed kdb Insights software. FinSpace manages and applies all such updates.

Some maintenance may require FinSpace to take your Managed kdb cluster offline for a short time. This includes, installing or upgrading required operating system or database patches. This maintenance is automatically scheduled for patches that are related to security and instance reliability.

The maintenance window determines when pending operations start, but it doesn't limit the total execution time of these operations. Maintenance operations that don't finish before the maintenance window ends can continue beyond the specified end time.

### Managed kdb Insights maintenance window

Every Manged kdb environment has a weekly maintenance window during which system changes are applied. You can control when modifications and software patches occurs during a maintenance window. If a maintenance event is scheduled for a given week, it is initiated during the maintenance window.

| AWS Region name | Time block |
|---|---|
| Canada (Central) | 15:00–16:30 UTC |
| US West (N. California) | 18:00–19:30 UTC |
| US West (Oregon) | 18:00–19:30 UTC |
| US East (N. Virginia) | 15:00–16:30 UTC |
| US East (Ohio) | 15:00–16:30 UTC |
| Europe (Ireland) | 10:00–11:30 UTC |
| Europe (London) | 09:00–10:30 UTC |
| Europe (Frankfurt) | 08:00–09:30 UTC |
| Asia Pacific (Singapore) | 02:00–03:30 UTC |
| Asia Pacific (Sydney) | 23:00–12:30 UTC |
| Asia Pacific (Tokyo) | 01:00–02:30 UTC |

## Creating a Managed kdb Insights cluster

You can either use the console or the CreateKxCluster API to create a cluster. When you create a cluster from the console, you choose one of the following cluster types available in FinSpace – General purpose, Tickerplant, HDB, RDB, and Gateway. The create cluster workflow includes a step-wise wizard, where you will add various details based on the cluster type you choose. The fields on each page can differ based on various selections throughout the cluster creation process.

### Prerequisites

Before you proceed, complete the following prerequisites:

- If you want to run clusters on a scaling group, create a scaling group.
- If you want to run a TP, GP, or RDB cluster on e scaling group create a volume.
- If you want to run an HDB type cluster on a scaling group, create a dataview.

**Topics**



### Opening the cluster wizard

**To open the create cluster wizard**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. In the kdb environments table, choose the name of the environment.
4. On the environment details page, choose **Clusters** tab.
5. Choose **Create cluster**. A step-wise wizard to create a cluster opens.

### Step 1: Add cluster details

Specify details for each of the following sections on **Add cluster details** page.

**Cluster details**

1. Choose from one of the following types of clusters that you want to add.

   - **(HDB) Historical Database**
   - **(RDB) Realtime Database**
   - **Gateway**
   - **General purpose**
   - **Tickerplant**

   For more information about cluster types, see Managed kdb Insights clusters.

> **Note**
>
> - Currently, you can only create dedicated HDB clusters and all cluster types that are running on a scaling group directly from the console. To create other types of clusters, you need to first create a Support case, and then proceed with steps in this tutorial.
> - The parameters that are displayed on the **Step 5: Configure data and storage** page will change based on the cluster type and running mode that you select in this step.

2. Add a unique name and a brief description for your cluster.
3. For `Release label`, choose the package version to run in the cluster.
4. (Optional) Choose the IAM role that defines a set of permissions associated with this cluster. This is an execution role that will be associated with the cluster. You can use this role to control access to other clusters in your Managed kdb environment.

**Cluster running mode**

1. Choose if you want to add this cluster as a dedicated cluster or as a part of scaling groups.
   - **Run on kdb scaling group** – Allows you to share a single set of compute with multiple clusters.
   - **Run as a dedicated cluster** – Allows you to run each process on its own compute hose.
2. If you choose **Run as a dedicated cluster**, you also need to provide the Availability Zones where you want to create a cluster.
   a. Choose **AZ mode** to specify the number of Availability Zones where you want to create a cluster. You can choose from one of the following options:
      - **Single** – Allows you to create a cluster in one Availability Zone that you select. If you choose this option, you must specify only one Availability Zone value and only one subnet in the next step. The subnet must reside in one of the three AZs that your kdb environment uses, and the Availability Zone must align with one of the three AZs.
      - **Multiple** – Allows you to create a cluster with nodes automatically allocated across all the Availability Zones that are used by your Managed kdb environment. This option provides resiliency for node or cache failures in a Single-AZ. If you choose this option,

you must specify three subnets, one in each of the three AZs that your kdb environment uses.

> **Note**
>
> For the **General purpose** and **Tickerplant** type cluster, you can only choose Single-AZ.

b. Choose the Availability Zone IDs that include the subnets you want to add.

**Scaling group details**

> **Note**
>
> This section is only available when you choose to add cluster as a part of scaling groups.

Choose the name of the scaling group where you want to create this cluster. The drop down shows the metadata for each scaling group along with their names to help you decide which one to pick. If a scaling group is not available, choose **Create kdb scaling group** to add a new one. For more information, see Creating a Managed kdb scaling group.

**Node details**

In this section, you can choose the capacity configuration for your clusters. The fields in this section vary for dedicated and scaling group clusters.

Scaling group cluster

You can decide the memory and CPU usage that will be shared with the instances for your scaling group clusters by providing the following information.

1. Under **Node details**, for **Node count**, enter the number of instances in a cluster.

> **Note**
>
> For a **General purpose** and **Tickerplant** type cluster, the node count is fixed at 1.

2. Enter the memory reservation and limits per node. Specifying the memory limit is optional. The memory limit should be equal to or greater than the memory reservation.
3. (Optional) Enter the number of vCPUs that you want to reserve for each node of this scaling group cluster.

Dedicated cluster

For a dedicated cluster you can provide an initial node count and choose the capacity configuration from a pre-defined list of node types. For example, the node type `kx.s.large` allows you to use two vCPUs and 12 GiB of memory for your instance.

1. Under **Node details**, for **Initial node count**, enter the number of instances in a cluster.

> **Note**
>
> For a **General purpose** and **Tickerplant** type cluster, the node count is fixed at 1.

2. For **Node type**, choose the memory and storage capabilities for your cluster instance. You can choose from one of the following options:

- `kx.s.large` – The node type with a configuration of 12 GiB memory and 2 vCPUs.
- `kx.s.xlarge` – The node type with a configuration of 27 GiB memory and 4 vCPUs.
- `kx.s.2xlarge` – The node type with a configuration of 54 GiB memory and 8 vCPUs.
- `kx.s.4xlarge` – The node type with a configuration of 108 GiB memory and 16 vCPUs.
- `kx.s.8xlarge` – The node type with a configuration of 216 GiB memory and 32 vCPUs.
- `kx.s.16xlarge` – The node type with a configuration of 432 GiB memory and 64 vCPUs.
- `kx.s.32xlarge` – The node type with a configuration of 864 GiB memory and 128 vCPUs.

**Auto-scaling**

> **Note**
>
> This section is only available when you add an HDB cluster type as a dedicated cluster.

Specify details to scale in or scale out the based on service utilization. For more information, see Auto scaling.

1. Enter a minimum node count. Valid numbers: 1–5.
2. Enter a maximum node count. Valid numbers: 1–5.
3. Choose the metrics to auto scale your cluster. Currently, FinSpace only supports CPU utilization.
4. Enter the cooldown time before initiating another scaling event.

**Tags**

1. (Optional) Add a new tag to assign to your kdb cluster. For more information, see AWS tags.

   > **Note**
   >
   > You can only add up to 50 tags to your cluster.

2. Choose **Next** for next step of the wizard.

**Step 2: Add code**

You can load q code onto your kdb cluster so that you can run it when the cluster is running. Additionally, you can configure your cluster to automatically run a particular q command script on cluster startup. By default, q writes files uncompressed. You can pass command line arguments to set compression defaults `.z.d` at the time of creating a cluster from the console or through CLI, which can be updated later.

> **Note**
>
> This step is required for the **Gateway** and **Tickerplant** cluster type.

On the **Add code** page, add the following details of your custom code that you want to use when analyzing the data in the cluster.

1. (Optional) Specify the **S3 URI** and the **Object version**. You can choose the *.zip* file that contains code that should be available on the cluster.

2. (Optional) For **Initialization script**, enter the relative path that contains a q program script that will run at the launch of a cluster. If you choose to load the database by using the initialization script, it will autoload on startup. If you add a changeset that has a missing sym file, the cluster creation fails.

   > **Note**
   >
   > This step is optional. If you choose to enter the initialization script, you must also provide the S3 URI.

3. (Optional) Enter key-value pairs as command-line arguments to configure the behavior of clusters. You can use the command-line arguments to set zip defaults for your clusters. For this, pass the following key-value pair:

   - **Key**: `AWS_ZIP_DEFAULT`
   - **Value**: `17,2,6`

     The value consists of comma separated three numbers that represent logical block size, algorithm, and compression level respectively. For more information, see compression parameters. You can also add the key-value pair when you update code configuration on a cluster.

   > **Note**
   >
   > You can only add up to 50 key-value pairs.

   To set compression default using AWS CLI, use the following command:

   ```
   aws finspace create-kx-cluster \
       ...
       --command-line-arguments '[{"key": "AWS_ZIP_DEFAULT", "value":"17,2,6"}]' \
       ...
   ```

4. Choose **Next**.

> **Note**
>
> In case of failure, to stop cluster creation from an initialization script, use the `.aws.stop_current_kx_cluster_creation` function in the script.

### Step 3: Configure VPC settings

You connect to your cluster using q IPC through an AWS PrivateLink VPC endpoint. The endpoint resides in a subnet that you specify in the AWS account where you created your Managed kdb environment. Each cluster that you create has its own AWS PrivateLink endpoint, with an elastic network interface that resides in the subnet you specify. You can specify a security group to be applied to the VPC endpoint.

Connect a cluster to a VPC in your account. On the **Configure VPC settings** page, do the following:

1. Choose the VPC that you want to access.
2. Choose the VPC subnets that the cluster will use to set up your VPC configuration.
3. Choose the security group.
4. Choose **Next**.

### Step 4: Configure data and storage

Choose data and storage configurations that will be used for the cluster.

The parameters on this page are displayed according to the cluster type that you selected in *Step 1: Add cluster details*.

> **Note**
>
> If you choose to add both the **Read data configuration** and **Savedown storage configuration**, the database name must be the same for both the configurations.

**For HDB cluster**

> **Note**
>
> When you create a cluster with a database that has a changeset, it will autoload the database when you launch a cluster.

If you choose **Cluster type** as *HDB*, you can specify the database and cache configurations as following:

Scaling group cluster

1. Choose the name of the database.
2. Choose a dataview for the database you selected.

   > **Note**
   >
   > If a dataview is not available in the list, either choose **Create dataview** to create a new one for the database you selected or try changing the availability zone.

3. Choose **Next**. The **Review and create** page opens.

Dedicated clusters

1. Choose the name of the database. This database must have a changeset added to it.
2. Choose the changeset that you want to use. By default, this field displays the most recent changeset.
3. Choose whether you want to cache your data from your database to this cluster. If you choose to enable caching, provide the following information:

   a. Choose the cache type, which is a type of read-only storage for storing a subset of your database content for faster read performance. You can choose from one of the following options:

   - **CACHE_1000** – Provides a throughput of 1000 MB/s per unit storage (TiB).
   - **CACHE_250** – Provides a throughput of 250 MB/s per unit storage (TiB).
   - **CACHE_12** – Provides a throughput of 12 MB/s per unit storage (TiB).

b. Choose the size of the cache. For cache type **CACHE_1000** and **CACHE_250** you can select cache size as 1200 GB or increments of 2400 GB. For cache type **CACHE_12** you can select the cache size in increments of 6000 GB.

4. Choose **Next**. The **Review and create** page opens.

**For RDB cluster**

If you choose **Cluster type** as *RDB*, you can specify the savedown storage configurations for your cluster as following:

Scaling group cluster

1. **Savedown database configuration**

   Choose the name of the database where you want to save your data.

2. **(Optional) Savedown storage configuration**

   Choose the name of the storage volume for your savedown files that you created in advance. If a volume name is not available, choose **Create volume** to create it.

3. **(Optional) Tickerplant log configuration**

   Choose a **Volume name** to use the tickerplant logs from.

4. Choose **Next**. The **Review and create** page opens.

Dedicated clusters

1. **Savedown database configuration**

   Choose the name of the database where you want to save your data.

2. **(Optional) Savedown storage configuration**

   1. Choose the writeable storage space type for temporarily storing your savedown data. Currently, only the **SDS01** storage type is available. This type represents 3000 IOPS and the Amazon EBS volume type `io2`.
   2. Enter the size of the savedown storage that will be available to the cluster in GiB.

3. **Tickerplant log configuration**

   Choose one or more volume names to use the tickerplant logs from.

4. Choose **Next**. The **Review and create** page opens.

**For Gateway cluster**

If you choose **Cluster type** as **Gateway**, you do not need to attach databases, cache configurations, or local storage in this step.

**For General purpose cluster**

If you choose **Cluster type** as *General purpose*, you can specify the database and cache configurations and savedown storage configurations as following:

Scaling group cluster

1. **(Optional) Read data configuration**

   1. Choose the name of the database.
   2. Choose a dataview for the database you selected.

   > **Note**
   >
   > If a dataview is not available in the list, either choose **Create dataview** to create a new one for the database you selected or try changing the availability zone.

2. **(Optional) Savedown database configuration**

   Choose the name of the database where you want to save your data.

3. **(Optional) Savedown storage configuration**

   Choose the name of the storage volume for your savedown files that you created in advance. If a volume name is not available, choose **Create volume** to create it.

4. **(Optional) Tickerplant log configuration**

   Choose a **Volume name** to use the tickerplant logs from.

5. Choose **Next**. The **Review and create** page opens.

Dedicated clusters

1. **(Optional) Read data configuration**

1. Choose the name of the database. This database must have a changeset added to it.
2. Choose the changeset that you want to use. By default, this field displays the most recent changeset.
3. Choose whether you want to cache your data from your database to this cluster. If you choose to enable caching, provide the following information:
   a. Specify paths within the database directory where you want to cache data.
   b. Choose the cache type, which is a type of read-only storage for storing a subset of your database content for faster read performance. You can choose from one of the following options:
      - **CACHE_1000** – Provides a throughput of 1000 MB/s per unit storage (TiB).
      - **CACHE_250** – Provides a throughput of 250 MB/s per unit storage (TiB).
      - **CACHE_12** – Provides a throughput of 12 MB/s per unit storage (TiB).
   c. Choose the size of the cache. For cache type **CACHE_1000** and **CACHE_250** you can select cache size as 1200 GB or increments of 2400 GB. For cache type **CACHE_12** you can select the cache size in increments of 6000 GB.

2. **(Optional) Savedown database configuration**

   Choose the name of the database where you want to save your data.

3. **(Optional) Savedown storage configuration**

   1. Choose the writeable storage space type for temporarily storing your savedown data. Currently, only the **SDS01** storage type is available. This type represents 3000 IOPS and the Amazon EBS volume type `io2`.
   2. Enter the size of the savedown storage that will be available to the cluster in GiB.

4. **Tickerplant log configuration**

   Choose one or more volume names to use the tickerplant logs from.

5. Choose **Next**. The **Review and create** page opens.

**For Tickerplant cluster**

For both scaling groups clusters and dedicated clusters, you can choose a volume where you want to store the tickerplant data.

1. **Tickerplant log configuration**

Choose a **Volume name** to store the tickerplant logs.

2. Choose **Next**. The **Review and create** page opens.

#### Step 5: Review and create

1. On the **Review and create** page, review the details that you provided. You can modify details for any step when you choose **Edit** on this page.
2. Choose **Create cluster**. The cluster details page opens where you can view the status of cluster creation.

## Viewing kdb cluster detail

#### To view and get details of a kdb cluster

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Clusters** tab. The table under this tab displays a list of clusters.
5. Choose a cluster name to view its details. The cluster details page opens where you can view the cluster details and the following tabs.

   - **Configuration** tab – Displays the cluster configuration details like the node details, code, availability zones, savedown database configuration etc.
   - **Monitoring** tab – Displays the dashboard of cluster metrics.
   - **Nodes** tab – Displays a list of nodes in this cluster along with their status. All the nodes that are active will have a **Running** status and nodes that are being prepared or stuck due to lack of resources have the status as **Provisioning**. From here you could also delete a node. For this, select a node and choose **Delete**.
   - **Logs** section – Displays the activity logs for your clusters.
   - **Tags** tab – Displays a list of key-value pairs associated with the clusters. If you did not provide tags during cluster creation, choose **Manage tags** to add new tags.

## Updating code configurations on a running cluster

Amazon FinSpace allows you to update code configurations on a running cluster. You can either use the console or the UpdateKxClusterCodeConfiguration API to update the code. Both console and API allow you to choose how you want to update the code on a cluster by using different deployment modes. Based on the option you choose, you can reduce the time it takes to update the code on to a cluster. You can also add or delete default compression parameters for your files by using command-line arguments.

> **Note**
>
> The configuration that you update will override any existing configurations on the cluster.

### To update code configurations on a cluster by using the console

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the list of environments, choose a kdb environment.
4. On the environment details page, choose the **Clusters** tab.
5. From the list of clusters, choose the one where you want to update the code. The cluster details page opens.
6. On the cluster details page, choose the **Details** tab.
7. Under **Code** section, choose **Edit**.

   > **Note**
   >
   > This button is only available for an **Active** environment and when the cluster is in a **Running** state.

8. On the **Edit code configuration** page, choose how you want to update a cluster by choosing a deployment mode. The following options are available.

   - **Rolling** – (Default) Loads the code by stopping the exiting q process and starting a new q process with updated configuration.
   - **Quick** – Loads the code by stopping all the running nodes immediately.

9. Specify the **S3 URI** and the **Object version**. This allows you to choose the *.zip* file containing code that should be available on the cluster.
10. For **Initialization script**, enter the relative path that contains a q program script that will run at the launch of a cluster.
11. (Optional) Add or update the key-value pairs as command line arguments to configure the behavior of clusters.

    You can use the command-line arguments to set zip defaults for your cluster. The cluster has to be restarted for the changes to take effect. For this, pass the following key-value pair:

    - **Key**: `AWS_ZIP_DEFAULT`
    - **Value**: `17,2,6`

      The value consists of comma separated three numbers that represent logical block size, algorithm, and compression level respectively. For more information, see compression parameters.

      To update the compression default using AWS CLI, use the following command:

      ```
      aws finspace update-kx-cluster-code-configuration \
          ...
          --command-line-arguments '[{"key": "AWS_ZIP_DEFAULT", "value":"17,3,0"}]' \
          --deploymentConfiguration deploymentStrategy=ROLLING|FORCE
          ...
      ```
12. Choose **Save changes**. The cluster details page opens and the updated code configuration is displayed once the cluster updates successfully.

## Updating a kdb cluster database

You can update the databases mounted on a kdb cluster using the console. This feature is only available for HDB clusters types. With this feature, you can update the data in a cluster by selecting a changeset. You can also update the cache by providing database paths. You can't change a database name or add a new database if you created a cluster without one.

You can also choose how you want to update the databases on the cluster by selecting a deployment mode.

**To update a kdb cluster database**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the list of environments, choose a kdb environment.
4. On the environment details page, choose the **Clusters** tab.
5. From the list of clusters, choose the one where you want to update the database. The cluster details page opens.
6. On the cluster details page, choose the **Details** tab.
7. Under **Data management and storage** section, choose **Edit**.

> **Note**
>
> This button is not available for *RDB* and *Gateway* type clusters.

8. On the edit page, modify the changeset that you want to cache as needed.
9. Choose a deployment mode from one of the following options.

   - **Rolling** – (Default) To update the database, this option stops the existing q process and starts a new q process with the updated database configuration. The initialization script re-runs when the new q process starts.
   - **No restart** – This option updates the database but doesn't stop the existing q process. **No restart** is often quicker than the other deployment modes because it reduces the turnaround time to update the changeset configuration for a kdb database on your cluster. This option doesn't re-run the initialization script.

> **Note**
>
> After the update completes, you must re-load the updated database. However, if you use a historical database (HDB) cluster with a single database in a rolling deployment, FinSpace autoloads the database after an update.

10. Choose **Save changes**. The cluster details page opens and the updated information is displayed once the cluster updates successfully.

### Deleting a kdb cluster

> **Note**
>
> This action is irreversible. Deleting a kdb cluster will delete all of its data from the local storage.

#### To delete a kdb cluster

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Clusters** tab.
5. From the list of clusters, choose the one that you want to delete. The cluster details page opens.
6. On the cluster details page, choose **Delete**.
7. On the confirmation dialog box, enter *confirm*.
8. Choose **Delete**.

#### Deleting a cluster on scaling groups

## Using Managed kdb Insights clusters

After you successfully create clusters in your kdb environment, you can use the clusters to do the following:

- **Monitor cluster metrics** – You can view the available cluster metrics in CloudWatch for your clusters. Using the **Monitoring** tab you can adjust the date and time range and refresh frequency as you need. You can also add the graphs to CloudWatch dashboards from this tab. For more information, see the Monitoring Managed kdb cluster metrics section.
- **View logs** – You can view KDB application logs from Managed kdb Insights clusters using the **Logs** tab. You can view data directly in CloudWatch using CloudWatch reporting or CloudWatch Insights. For more information, see the Logging section.

- **Connect to clusters** – FinSpace provides you the ability to discover clusters in your dedicated account and connect to them. You can do this by using discovery API operations and q API operations. For more information on how to connect to a cluster, see the Connecting to a cluster endpoint or node in a cluster section.
- **Load code on to a cluster** – You can run your own KDB code on the cluster and perform analytics or query data in a database. For this, FinSpace provides a set of q API operations that you can use to perform required functions. For more information, see the Running code on a Managed kdb Insights cluster section.

## Managing kdb users

The following sections provide a detailed overview of the operations that you can perform by using Managed kdb Insights users. A kdb user is required in order to establish a connection to a Managed kdb cluster. For more information, see Interacting with a kdb cluster.

### Creating a kdb user

**To create a user**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Users** tab.
5. Choose **Add user**.
6. On the **Add user** page, a unique name for the user.
7. Choose an IAM role available in your account to associate it with the user. This role will be used later when you connect to a cluster.

   > **Note**
   >
   > The IAM role that you choose must have connect cluster permissions.

8. (Optional) Add a new tag to assign it to your kdb user. For more information, see AWS tags.

> **Note**
>
> You can only add up to 50 tags to your user.

9. Choose **Add user**. The environment details page opens and the table under **Users** lists the newly added user.

### Updating a kdb user

> **Note**
>
> You can only modify the IAM role associated with a user.

### To update a kdb user

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Users** tab.
5. From the list of users, choose the one that you want to update.
6. Choose **Edit**.
7. Choose a new IAM role to associate with this user.
8. Choose **Update user**.

### Deleting a kdb user

> **Note**
>
> This action is irreversible.

**To delete a kdb user**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Users** tab.
5. From the list of users, choose the one that you want to delete.
6. Choose **Delete**.
7. On the confirmation dialog box, enter *confirm*.
8. Choose **Delete**.

## Interacting with a kdb cluster

To run commands on a Managed kdb Insights cluster, you must establish a q connection to a cluster endpoint or individual node in a cluster. If you don't care which node in the cluster your connection is established with, use the cluster endpoint. The endpoint is an IP address (elastic network interface) that resides in your account. This provides a simple way to connect for a single-node cluster and for other scenarios.

Alternately, from client code residing on a cluster node running with Managed kdb, you can also make a direct connection to an individual node. This gives you full control of which node in a cluster to use. This might be useful if you have custom allocation logic in your client code. You can use the Managed kdb list clusters and list node functionality to see what cluster and node resources are available in your environment. Then, you can use the cluster connection functionality to obtain a connection string that you can use to establish a q IPC connection to a cluster or node.

As a part of cluster discovery, FinSpace provides you the following capabilities:

- Listing all clusters running in your Managed kdb environment.
- Listing all nodes running in a kdb cluster. For more information on this, see Listing clusters and cluster nodes.
- Connect to the underlying node from an existing cluster. For more information on this, see Connecting to a cluster endpoint or node in a cluster.

### Listing clusters and cluster nodes

There are three ways to view a list of clusters and nodes running in a cluster:

- **FinSpace API operations** – You can call the `ListKxClusterNodes` API operation to get a list of nodes in a cluster. For more information, see the ListKxClusterNodes in the *Management API Reference Guide*.
- **q API operations** – You can use the `.aws.list_kx_cluster_nodes()` and `.aws.list_kx_clusters()` API operations to get a list of nodes or clusters. For more information, see Discovery APIs.
- **Console** – You can view a list of nodes from the cluster details page in the FinSpace console.

#### To view a list of clusters by using the console

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Clusters** tab.
5. Choose a cluster name to view its details. On the cluster details page, you can see details about a cluster.

#### To view a list of nodes available in a cluster by using the console

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Kdb environments**.
3. From the list of environments, choose a kdb environment.
4. On the environment details page, choose the **Clusters** tab.
5. From the list of clusters, choose the one where you want to view nodes.
6. On the cluster details page, choose **Nodes** tab. All the nodes running in the cluster are displayed along with the information about the node ID , the Availability Zone ID where the node is running, and the time when the node was started. You can use the `nodeId` to call the `GetKxConnectionString` API operation, which returns a signed connection string.

### Connecting to a cluster endpoint or node in a cluster

Amazon FinSpace uses the model based on AWS Identity and Access Management that allows users to control access to clusters and their associated kdb databases using IAM roles and policies.

Administrators create users in the FinSpace kdb environment using the existing `CreateKxUser` API operation, and associate these users with an IAM principal. Only users that will be connecting to a kdb cluster need to be registered as a FinSpace user.

Next, using their IAM credentials, connecting users will request a SigV4 signed authentication token to connect to the cluster. Additionally, each cluster can be associated with an IAM execution role in the customer account when a cluster is created. This role will be used when a cluster connects to other clusters, or makes requests to other AWS resources in the customer's account.

**To connect to a cluster endpoint or cluster node**

1. **Create IAM role for a new user.**

   a. Sign in to AWS Management Console, and open IAM Identity Center.

   b. Create an IAM role.

   c. Assign the following policy to the IAM role that you created.

      In the following example, replace each *`user input placeholder`* with your own values.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "finspace:ConnectKxCluster",
            "Resource": "arn:aws:finspace:us-east-1:111122223333:kxEnvironment/
sdb3moagybykax4oexvsq4/kxCluster/testhdb-cluster"
        },
        {
            "Effect": "Allow",
            "Action": "finspace:GetKxConnectionString",
            "Resource": "arn:aws:finspace:us-
east-1:111122223333:kxEnvironment/sdb3moagybykax4oexvsq4/kxCluster/testhdb-
cluster"
        }
    ]
```

```
}
```

d. Associate the IAM role to the following trust policy that allows FinSpace to assume the role, as well as the account itself.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "finspace.amazonaws.com",
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. **Create a kdb user with the environment id, username, and the IAM role that you created in the previous step.**

```
aws finspace create-kx-user
    --environment-id "sdb3moagybykax4oexvsq4"
    --user-name alice
    --iam-role arn:aws:iam::111122223333:role/user-alice
    [--tags {tags}]
```

3. **Federate the user that you created into its user role.**

a. To get a kdb connection string for a user, you must first federated into the role associated with the user. How you assume this role depends on what federation tool you use. If you use AWS Security Token Service, you could run the following command and use the credentials of the customer account.

```
export $(printf "AWS_ACCESS_KEY_ID=%s AWS_SECRET_ACCESS_KEY=%s
 AWS_SESSION_TOKEN=%s" \
$(aws sts assume-role \
--role-arn arn:aws:iam::111122223333:role/user-alice \
--role-session-name "alice-connect-to-testhdb" \
--query "Credentials.[AccessKeyId,SecretAccessKey,SessionToken]" \
--output text))
```

b. Verify that the role has been assumed.

```
aws sts get-caller-identity | cat
```

4. **Get connection string for the user.**

   Get signed connection strings for connecting to kdb clusters or nodes. These connection strings are valid only for 60 minutes. To connect to a cluster endpoint, use `get-kx-connetion-string` to obtain a connection string.

```
aws finspace get-kx-connection-string
    --environment-id "sdb3moax4oexvsq4"
    --user-arn arn:aws:finspace:us-
east-1:111122223333:kxEnvironment/sdb3moax4oexvsq4/kxUser/alice
    --cluster-name "testhdb-cluster"
    --region us-east-1
```

   Example of the signed connection string that you get.

```
:tcps://vpce-06259327736e61c9d-uczv1va3.vpce-svc-0938de45abc1ce4d8.us-
east-1.vpce.amazonaws.com:443:testuser:Host=vpce-06259327736e61c9d-
uczv1va3.vpce-svc-0938de45abc1ce4d8.us-
east-1.vpce.amazonaws.com&Port=5000&User=testuser&Action=finspace
%3AConnectKxCluster&X-Amz-Security-Token=IQoJb3JpZ2luX2Vj&X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20230524T150227Z&X-Amz-
SignedHeaders=host&X-Amz-Expires=900&X-Amz-Credential=ASIAR2V4%2Fus-
east-1%2Ffinspace-apricot%2Faws4_request&X-Amz-
Signature=28854cc2f97f8f77009928fcdf15480dd10b43c61dda22b0af5f0985d38e7114
```

5. Connect to a cluster using the signed connection string.

```
hopen :tcps://vpce-06259327736e61c9d-uczv1va3.vpce-svc-0938de45abc1ce4d8.us-
east-1.vpce.amazonaws.com:443:testuser:Host=vpce-06259327736e61c9d-uczv1va3.vpce-
svc-0938de45abc1ce4d8.us-
east-1.vpce.amazonaws.com&Port=5000&User=testuser&Action=finspace
%3AConnectKxCluster&X-Amz-Security-Token=IQoJb3JpZ2luX2Vj&X-Amz-Algorithm=AWS4-
HMAC-SHA256&X-Amz-Date=20230524T150227Z&X-Amz-SignedHeaders=host&X-Amz-
Expires=900&X-Amz-Credential=ASIAR2V4%2Fus-east-1%2Ffinspace-apricot
%2Faws4_request&X-Amz-
Signature=28854cc2f97f8f77009928fcdf15480dd10b43c61dda22b0af5f0985d38e7114
```

**Note**

The connection handles to the cluster VPC endpoint have an idle timeout period of 350 seconds. If you don't send commands or data by the time that the idle timeout period elapses, the connection closes and you will need to reopen it.
To keep the connection open, use a timer that periodically sends a ping message to the cluster through an active handle. For this, you can run the following code.

```
.aws.start_keepalive:
{\t 10000;.z.ts:{con "-1 \"Ping\"";}}
.aws.stop_keepalive:
{\t 0;.z.ts:{}}
```

## Running code on a Managed kdb Insights cluster

Q is the programming system for working with kdb+. This corresponds to SQL for traditional databases, but unlike SQL, q is a powerful interpreted programming language in its own right. Q expressions can be entered and run in the q console, or loaded from a q script, which is a text file with extension *.q*. For more information, see the q documentation. You can run your own kdb code on a Managed kdb cluster to perform analytics or query data in a database.

The following sections describe how you can use q in FinSpace.

### Topics

- .z namespace override
- Supported system commands
- FinSpace q API reference

### .z namespace override

KX uses the `.z` namespace that contains environment variables and functions, and hooks for callbacks. A FinSpace Managed kdb Insights cluster doesn't support direct assignment for `.z` namespace callbacks because of security concerns. For example, the system denies access to the following direct `.z.ts` assignment.

```
q)con".z.ts:{[x]}" / con is the hopen filehandle
'access
```

```
[0]  con".z.ts:{[x]}"
```

Because some of the assignments for `.z` namespace callbacks are critical for business logic, FinSpace provides a reserved namespace `.awscust.z` for you to override functions within the `.z` namespace.

By overriding the functions within the new `.awscust.z` namespace, you can achieve the same effect as if you were directly overriding allowlisted `.z` functions.

For example, if you need to override the `.z.ts` function, you can set a value for `.awscust.z.ts`. The FinSpace Managed kdb cluster invokes the `.awscust.z.ts` function whenever you invoke the `.z.ts` function, which provides a safety wrapper.

The following list shows the allowlisted callbacks for the `.awscust.z` namespace.

```
.awscust.z.ts
.awscust.z.pg
.awscust.z.ps
.awscust.z.po
.awscust.z.pc
.awscust.z.ws
.awscust.z.wo
.awscust.z.wc
.awscust.z.pd
.awscust.z.ph
.awscust.z.pi:
.awscust.z.exit
```

If you override `.z` callbacks that aren't on the preceding list, you won't have any effects on the `.z` namespace callbacks.

**Supported system commands**

System commands control the q environment. The following table shows a list of system commands that FinSpace supports.

| System commands | Description | Constraints |
|---|---|---|
| \a | Lists all the tables in the current namespace. | None |

| System commands | Description | Constraints |
|---|---|---|
| `\awk` | A pattern scanning and text processing language. | Only available on General purpose and HDB clusters. |
| `\b` | Lists all the views (derived tables). | None |
| `\B` | Lists all the pending views. | |
| `\base64` | Encodes or decodes data in base64 format. | Only available on General purpose and HDB clusters. |
| `\c` | Shows or sets the console size. | None |
| `\C` | Shows or sets the HTTP response size. | |
| `\cat` | Concatenates and display files. | Only available on General purpose and HDB clusters. |
| `\cd` | Shows or sets the current directory. | |
| `\cp` | Copies files or directories. | |
| `\curl` | Transfers data to or from a web server using HTTP, HTTPS, SCP, SFTP, TFTP, and more. | |
| `\d` | Changes the current namespace. | None |
| `\dirname` | Removes the last component of a file name, leaving the directory path. | Only available on General purpose and HDB clusters. |

| System commands | Description | Constraints |
| --- | --- | --- |
| `\e` | Governs error trapping for client requests. | None |
| `\echo` | Displays text or variables to the standard output. | Only available on General purpose and HDB clusters. |
| `\egrep` | Searches for a pattern in one or more input files (extended grep). | |
| `\f` | Lists all functions in the current namespace. | None |
| `\find` | Searches for files based on various criteria such as name, size, and modification time. | Only available on General purpose and HDB clusters. |
| `\g` | Shows or sets garbage-collection mode. | None |
| `\grep` | Looks for a pattern in one or more input files. | Only available on General purpose and HDB clusters. |
| `\gunzip` | Uncompresses a file or list of files. | |
| `\gzip` | Compresses a file or list of files. | |
| `\jq` | A lightweight and flexible command-line JSON processor. | |
| `\l` | Loads a script or data from a file or directory. | None |

| System commands | Description | Constraints |
| --- | --- | --- |
| `\ln` | Creates a hard link or symbolic link to a file. | Only available on General purpose and HDB clusters. |
| `\ls` | Lists information about files and directories. | |
| `\mkdir` | Creates a new directory. | |
| `\mv` | Moves or renames files or directories. | |
| `\nohup` | Runs a command immune to hangups, allowing it to continue running in the background. | |
| `\o` | Shows or sets the offset from Coordinated Universal Time (UTC). | None |
| `\p` | Shows or sets the TCP port on which the q session listens. | FinSpace supports this command with no arguments , which gets the current port. It is only permitted with argument 443 (noop), not permitted with other arguments. |
| `\P` | Sets the display precision for floating-point numbers. | None |
| `\pgrep` | Looks up or signal processes based on name and other attributes. | Only available on General purpose and HDB clusters. |
| `\ps` | Displays information about running processes. | |

| System commands | Description | Constraints |
|---|---|---|
| `\pwd` | Prints the current working directory. | |
| `\r` | Renames a file. | None |
| `\readlink` | Displays the value of a symbolic link. | Only available on General purpose and HDB clusters. |
| `\rm` | Removes files or directories. | |
| `\rmdir` | Removes empty directories. | |
| `\s` | Shows or sets the number of secondary threads available. | None |
| `\S` | Shows or sets the value of the random seed. | |
| `\sed` | A stream editor that filters and transforms text. | Only available on General purpose and HDB clusters. |
| `\sleep` | Suspends execution for a specified period of time. | |
| `\t` | Shows or sets the timer interrupt in milliseconds. | FinSpace supports this command but it is not fully functional as the function called by the timer can only be set by the init script. |
| `\T` | Shows or sets the client execution timeout. | None |
| `\touch` | Creates a new file or updates the timestamp of an existing file. | Only available on General purpose and HDB clusters. |

| System commands | Description | Constraints |
|---|---|---|
| `\tr` | Translates or deletes characters from standard input. | |
| `\ts` | Runs an expression and shows the runtime and memory used. | None |
| `\unzip` | Extracts files from a ZIP archive. | Only available on General purpose and HDB clusters. |
| `\v` | Lists variables in the current or specified namespace. | None |
| `\w` | Shows memory usage or sets workspace memory limit. | |
| `\W` | Shows or sets the start-of-week offset. | |
| `\wc` | Counts the number of lines, words, and characters in one or more files. | Only available on General purpose and HDB clusters. |
| `\xargs` | Builds and executes command lines from standard input. | |
| `\z` | Shows or sets the format for date parsing. | None |
| `\zip` | Packages and compresses files into a ZIP archive. | Only available on General purpose and HDB clusters. |
| `\_` | In debugger's prompt, clears one level from the execution stack or toggles between the q and k interpreters. | None |

| System commands | Description | Constraints |
| --- | --- | --- |
| \1 | Redirects stdout to files from within the q session. | |
| \2 | Redirects stderr to files from within the q session. | |

### Helper environment variables

You can quickly access user directories through the following environment variables that return a string of the folder path.

| Helper environment variables | Use for | Directory |
| --- | --- | --- |
| `.aws.akcp` | Primary user code path. | `/opt/kx/app/code` |
| `.aws.akcsp` | Secondary user code path that's available only for **General purpose** cluster. | `/opt/kx/app/code_scratch` |
| `.aws.akscp` | Primarily used for handling savedown functionality with an RDB cluster. | `/opt/kx/app/scratch` |

### Loading databases relative to code directory

We have added a symlink to the code directory to allow loading of database relative to the code path. For example, if the database is labeled as *kxDatabase* and the current working directory is `/opt/kx/app/code` then the database can be loaded as `\l /kxDatabase`.

### FinSpace q API reference

FinSpace provides a set of q APIs that you can use to interact with resources in your Managed kdb environment. These APIs reside in the *.aws Q* namespace.

**Ingestion APIs**

**Function**: `.aws.create_changeset[db_name;change_requests]`

Creates a new changeset in the specified database.

Parameters

`db_name`

**Description** – The name of the FinSpace kdb database where you can create Managed kdb Insights changesets. This must be the same database that you used when you created the RDB cluster.

**Type** – String

**Required** – Yes

`change_requests`

**Description** – A q table representing the list of change requests for the Managed kdb Insights changesets. The table has 3 columns :

- `input_path` – The input path of the local file system directory or file to ingest as a Managed kdb changeset.
- `database_path` – The target database destination path of the Managed kdb changeset. This column maps to the `databasePath` field of the CreateKxChangeset API.
- `change_type` – The type of the Managed kdb changeset. It can be either `PUT` or `DELETE`. This column maps to the `changeType` field of the CreateKxChangeset API.

**Type** – Q table

**Required** – Yes

Result

Returns the `changeset_id` of the created Managed kdb changeset, along with its current status.

**Function**: `.aws.get_changeset[db_name;changeset_id]`

Retrieves information about a specific changeset.

Parameters

`db_name`

**Description** – The name of the FinSpace kdb database where you can create Managed kdb changesets. This must be the same database that you used when you created the RDB cluster.

**Type** – String

**Required** – Yes

`changeset_id`

**Description** – The identifier of the Managed kdb changeset.

**Type** – string

**Required** – Yes

Result

Returns the `changeset_id` and the status of the Managed kdb changeset.

**Function**: `.aws.get_latest_sym_file[db_name;destination_path]`

Retrieves the latest sym file from the specified database.

Parameters

`db_name`

**Description** – The name of the FinSpace kdb database where you can create Managed kdb changesets. This must be the same database that you used when you created the RDB cluster.

**Type** – String

**Required** – Yes

`destination_path`

**Description** – The directory in the local filesystem scratch location where you want to download the symfile.

**Type** – String

**Required** – Yes

Result

Returns the destination path where the sym file was copied to.

**Function**: `.aws.s3.get_object[source_s3_path;destination_disk_path]`

Copies Amazon S3 object from your S3 bucket account into a local disk location in a kdb cluster.

Permissions

For this function, the `executionRole` of the cluster must have the `s3:GetObject` permission to access the object and `kms:Decrypt` permission for the key that you use to encrypt the S3 bucket.

Parameters

`source_s3_path`

**Description** – The source path in the customer account from where you want to copy an S3 object. This can be S3 object ARN or S3 URI path.

**Type** – String

**Required** – Yes

`destination_disk_path`

**Description** – The local disk location to copy the S3 object to.

**Type** – String

**Required** – Yes

Example

The following code is an example request to copy S3 object to a local disk.

```
 q) .aws.s3.get_object["s3://customer-bucket/reference_data.csv"; "/opt/kx/app/
shared/VolumeName/common/"]
```

Result

Returns a table of S3 object path and local directory disk location.

Sample response

```
s3ObjectPath                                containerFileDestinationPath
-------------------------------------------------------------------------
s3://data-bucket/data.csv                   "/opt/kx/app/shared/test/common/
data.csv"
```

Sample response for retrieving multiple files

```
.aws.copy_database_files["DATABASE_NAME"; "DESTINATION_PATH"; "PARTITION_NAME/*";
 ""]
database_name| "DATABASE_NAME"
changeset_id | "CHANGESET_ID"
result_paths | ("DESTINATION_PATH/PARTITION_NAME/file1"; "DESTINATION_PATH/
PARTITION_NAME/file2"...)
```

**Function**: `.aws.copy_database_files[database_name, destination_path, db_path, changeset_id]`

Retrieves a specific file from a specific version of the database. The `changeset_id` provides the version of the database from where you want to retrieve the file.

Parameters

`database_name`

**Description** – The name of the FinSpace kdb database where you can create Managed kdb changesets. This must be the same database that you used when you created the RDB cluster.

**Type** – String

**Required** – Yes

`destination_path`

**Description** – The directory in the local filesystem scratch location where you want to download one or more files.

**Type** – String

**Required** – Yes

`db_path`

**Description** – The path within the database directory of the file you want to retrieve. This can be a single file or a path ending with the wildcard "*" to retrieve multiple files. Following are a few example values for `db_path`.

- `sym` retrieves the file named **sym** located in the root directory of the database.
- `sym*` retrieves all files starting with **sym** for a database. For example, **sym1** and **sym2**.
- `2022.01.02/*` retrieves all files within the directory **2022.01.02**. For example, **2022.01.02/col1**, **2022.01.02/col2**, etc. Alternatively, you can use `2022.01.02/` to achieve the same result.
- `2022.05.*` retrieves all files from May 2022 within a date-partitioned database. For example, all files from **2022.05.01**, **2022.05.02**, etc.

**Type** – String

**Required** – Yes

`changeset_id`

**Description** – The identifier of the Managed kdb changeset. You can specify an empty string `""` to use the latest changeset.

**Type** – String

**Required** – Yes

Result

Returns the destination path where the files were copied to, along with the `database_name` and `changeset_id` used.

Sample response for retrieving a file

```
.aws.copy_database_files["DATABASE_NAME"; "DESTINATION_PATH"; "DB_FILE_PATH"; ""]
database_name| "DATABASE_NAME"
changeset_id | "CHANGESET_ID"
result_paths | ,"DESTINATION_PATH/DB_FILE_PATH"
```

Sample response for retrieving multiple files

```
.aws.copy_database_files["DATABASE_NAME"; "DESTINATION_PATH"; "PARTITION_NAME/*";
 ""]
database_name| "DATABASE_NAME"
changeset_id | "CHANGESET_ID"
result_paths | ("DESTINATION_PATH/PARTITION_NAME/file1"; "DESTINATION_PATH/
PARTITION_NAME/file2"...)
```

**Function**: `.aws.get_kx_dataview[db_name;dataview_name]`

Retrieves information about a specific dataview. This operation is helpful especially when you update a dataview with `update_kx_dataview`, as it retrieves the latest status and reflects the updated `changeset_id`, `segment_configurations`, and `active_versions`.

Permissions

For this function, the `executionRole` must have the `finspace:GetKxDataview` permission.

Parameters

`db_name`

**Description** – The name of the FinSpace kdb database where the specified dataview exists. This must be same as the database you used when you created a cluster.

**Type** – String

**Required** – Yes

`dataview_name`

**Description** – The name of the Managed kdb dataview you want to retrieve.

**Type** – String

**Required** – Yes

Result

Returns the details of specified dataview, including its status, changeset id, and segment configurations.

```
dataview_name          | "example-dataview-name"
database_name          | "example-db"
status                 | "ACTIVE"
changeset_id           | "example-changeset-id"
segment_configurations | +`db_paths`volume_name!(,,"/*";,"example-volume")
availability_zone_id   | "use1-az2"
az_mode                | "SINGLE"
auto_update            | 0b
read_write             | 0b
active_versions        |
 +`changeset_id`segment_configurations`attached_clusters`created_timestamp`version_id!
(("example-changeset-id";"prior-changeset-id");(+`db_paths`volume_name!
(,,"/*";,"example-volume");+`db_paths`volume_name!(,,"/*";,"example-
volume"));();,"example-cluster");1.717532e+09 1.716324e+09;
("kMfybotBQNQl5LBLhDnAEA";"XMfOcGisErAFO9i1XRTdYQ"))
create_timestamp       | 1.716324e+09
last_modified_timestamp| 1.717779e+09
```

**Function**:
`.aws.update_kx_dataview[db_name;dataview_name;changeset_id;segment_configuration`

Updates the changeset id and/or segment configurations of the specified dataview. Each update of the dataview creates a new version, with its own changeset details and cache configurations. If a dataview is created with auto-update set to false, when new changesets are ingested, this operation must be run to update the dataview with the latest changeset. This operation can also be used to update the segment configurations, which define which database paths are placed on each selected volume.

Permissions

For this function, the `executionRole` must have the `finspace:UpdateKxDataview` permission.

Parameters

db_name

**Description** – The name of the Managed kdb database where the specified dataview exists. This must be the same database that you used when you created a cluster.

**Type** – String

**Required** – Yes

`dataview_name`

**Description** – The name of the Managed kdb dataview you want to update.

**Type** – String

**Required** – Yes

`changeset_id`

**Description** – The identifier of the Managed kdb changeset that the dataview should use.

**Type** – String

**Required** – Yes

`segment_configurations`

**Description** – The output of the `.aws.sgmtcfgs` function.

**Required** – Yes

Example

The following code is an example request to update the dataview.

```
.aws.update_kx_dataview["example-db"; "example-dataview-name"; "example-changeset-
id"; .aws.sgmtcfgs[.aws.sgmtcfg[("/*");"example-volume"]]]
```

Result

This function does not return any value.

**Function**: `.aws.sgmtcfgs[segment_configurations]`

This is a helper function to construct arguments for `.aws.update_kx_dataview`, defining the list of segment configurations for the dataview.

Parameters

`segment_configurations`

**Description** – Either a single output of `.aws.sgmtcfg` or a list of `.aws.sgmtcfg` outputs.

**Required** – Yes

Example

The following example shows how this function can take a single segment configuration.

```
.aws.sgmtcfgs[.aws.sgmtcfg[("/*");"example-volume"]]
```

Alternatively, you can use this function with multiple segment configurations as following.

```
.aws.sgmtcfgs[(.aws.sgmtcfg[("/2020.02.01/*");"example-
volume-1"];.aws.sgmtcfg[("/2020.02.02/*");"example-volume-2"])]
```

Result

The output of this function is used as input for `.aws.update_kx_dataview`.

**Function**: `.aws.sgmtcfg[db_paths;volume_name]`

This is a helper function to construct arguments for `.aws.sgmtcfgs`, defining a single segment configuration, the database path of the data that you want to place on each selected volume. Each segment must have a unique database path for each volume.

Parameters

`db_paths`

**Description** – The database path of the data you want to place on each selected volume for the segment. Each segment must have a unique database path for each volume.

**Type** – Array of strings

**Required** – Yes

`volume_name`

**Description** – The name of the Managed kdb volume where you would like to add data.

**Type** – String

**Required** – Yes

Example

The following example shows how this function can take a single db path.

```
.aws.sgmtcfg[("/*");"example-volume"]
```

Alternatively, you can use this function with multiple db paths as following.

```
.aws.sgmtcfg[("/2020.01.06/*";"/2020.01.02/*");"example-volume"]
```

Result

The output of this function is used as input for `.aws.sgmtcfgs`.

**Discovery APIs**

**Function**: `.aws.list_kx_clusters()`

Returns a table of clusters in non-deleted state.

Parameters

N/A

Result

Returns a table of Managed kdb clusters that are in a non-deleted state. This table consists of the following fields – `cluster_name`, `status`, `cluster_type`, and `description`.

**Function**: `.aws.list_kx_cluster_nodes[cluster_name]`

Retrieves a list of nodes within a cluster.

Parameters

`cluster_name`

**Description** – The name of the Managed kdb cluster that you specified when creating a kdb cluster. You can also get this by using the `.aws.list_kx_clusters()` function.

**Type** – String

**Required** – Yes

Result

Returns a table of nodes in the Managed kdb cluster that consists of `node_id`, `az_id`, and `launch_time`.

### Authorization APIs

> **Note**
>
> You must create clusters with the IAM `executionRole` field to use the q auth APIs. Clusters will assume this role when calling the auth APIs, so the role should have `GetKxConnectionString` and `ConnectKxCluster` permissions.

**Function**: `.aws.get_kx_node_connection_string[cluster_name;node_id]`

Retrieves the connection string for a given kdb cluster node.

Parameters

`cluster_name`

**Description** – The name of the destination Managed kdb cluster for the connection string.

**Type** – String

**Pattern** – `^[a-zA-Z0-9][a-zA-Z0-9-_]*[a-zA-Z0-9]$`

**Length** – 3-63

**Required** – Yes

`node_id`

**Description** – The node identifier of the target cluster.

**Type** – String

**Length** – 1-40

**Required** – Yes

Result

Returns the connection string.

**Function**: `.aws.get_kx_connection_string[cluster_name]`

Retrieves the connection string for a given kdb cluster.

Parameters

`cluster_name`

**Description** – The name of the destination cluster for the connection string.

**Type** – String

**Pattern** – `^[a-zA-Z0-9][a-zA-Z0-9-_]*[a-zA-Z0-9]$`

**Length** – 3-63

**Required** – Yes

Result

Returns the connection string.

### Cluster management APIs

> **Note**
>
> You must create clusters with the IAM `executionRole` field to use the cluster management APIs.

**Function**: `.aws.stop_current_kx_cluster_creation[message]`

Stops the current cluster creation and puts the cluster in the `CREATE_FAILED` state. You can only call this function from an initialization script.

Parameters

`message`

**Description** – A message to display in the `statusReason` field of the cluster after the cluster reaches the `CREATE_FAILED` state.

**Type** – String

**Pattern** – `^[a-zA-Z0-9\_\-\.\s]*$`

**Length** – 0-50

**Required** – Yes

Example

The following code is an example request to stop creation of current cluster with a message.

```
.aws.stop_current_kx_cluster_creation[""]
```

Result

This function does not return any value.

**Function**: `.aws.delete_kx_cluster[clusterName]`

Deletes the specified cluster. If `clusterName` is an empty string, this function deletes the current cluster.

Permissions

For this function, the `executionRole` must have the following permissions to delete the cluster:

- `ec2:DescribeTags`
- `ec2:DeleteVpcEndpoints`
- `finspace:DeleteKxCluster`

Parameters

`clusterName`

**Description** – The name of the cluster that you want to delete.

**Type** – String

**Pattern** – `^[a-zA-Z0-9-_]*$`

**Length** – 1-63

**Required** – Yes

Example

The following example deletes the *samplecst* cluster.

```
.aws.delete_kx_cluster["samplecst"]
```

The following example deletes the current cluster.

```
.aws.delete_kx_cluster[""]
```

Result

This function does not return any value.

**Function**: `.aws.get_kx_cluster[clusterName]`

Retrieves information about the specified cluster.

Permissions

For this function, the `executionRole` must have the `finspace:GetKxCluster` permission.

Parameters

`clusterName`

**Description** – The name of the target cluster.

**Type** – String

**Pattern** – `^[a-zA-Z0-9-_]*$`

**Length** – 1-63

**Required** – Yes

Result

```
status               | "RUNNING"
>>>>>>> mainline
clusterName          | "example-cluster-name"
clusterType          | "HDB"
capacityConfiguration| `nodeType`nodeCount!("kx.s.xlarge";1f)
releaseLabel         | "1.0"
vpcConfiguration     | `vpcId`securityGroupIds`subnetIds`ipAddressType!
("vpcId";,"securityGroupId";,"subnetId";"IP_V4")
executionRole        | "arn:aws:iam::111111111111:role/exampleRole"
lastModifiedTimestamp| 1.695064e+09
azMode               | "SINGLE"
availabilityZoneId   | "use1-az1"
createdTimestamp     | 1.695063e+09
```

**Function**: `.aws.update_kx_cluster_databases[clusterName;databases;properties]`

Updates the database of the specified kdb cluster.

Permissions

- For this function, the `executionRole` must have the `finspace:UpdateKxClusterDatabases` permission.
- You must have `finspace:GetKXCluster` permission for the `clusterName`.

Parameters

`clusterName`

**Description** – The name of the target cluster.

**Type** – String

**Pattern** – `^[a-zA-Z0-9][a-zA-Z0-9-_]*[a-zA-Z0-9]$`

**Length** – 3-63

**Required** – Yes

`databases`

**Description** – The output of the `.aws.sdbs` function.

**Required** – Yes

`properties`

**Description** – The output of the `.aws.sdep` function.

**Required** – Yes

Example

The following code is an example request to update the cluster database.

```
.aws.update_kx_cluster_databases["HDB_TAQ_2021H1";
    .aws.sdbs[
        .aws.db["TAQ_2021H1";"osSoXB58eSXuDXLZFTCHyg";
            .aws.cache["CACHE_1000";"/"];
            ""
            ]
        ];
    .aws.sdep["ROLLING"]]
```

Result

This function does not return any value.

**Function**: `.aws.sdbs[databases]`

This is a helper function to construct arguments for `.aws.update_kx_cluster_databases`.

Parameters

databases

**Description** – It is either a single output of `.aws.db` or a list of `.aws.db` outputs.

**Required** – Yes

Example

Here is an example of how you can use this function.

```
.aws.sdbs[
    .aws.db["TAQ_2021H1";"osSoXB58eSXuDXLZFTCHyg";
        .aws.cache["CACHE_1000";"/"];
         ""
        ]
    ];
```

Result

The output of this function is used as input for `.aws.update_kx_cluster_databases` function.

**Function**: `.aws.db[databaseName;changesetId;caches;dataviewName]`

This is a helper function to construct arguments for `.aws.sdbs`.

Parameters

databaseName

**Description** – The name of the target database.

**Type** – String

**Pattern** – `^[a-zA-Z0-9][a-zA-Z0-9-_]*[a-zA-Z0-9]$`

**Length** – 3-63

**Required** – Yes

changesetId

**Description** – A unique identifier of the changeset. If you pass empty string “” for this parameter, the latest changeset of the database will be used.

**Type** – String

**Length** – 1-26

**Required** – No

caches

**Description** – It is either a single output of `.aws.cache` or a list of .`aws.cache` outputs. If there is no cache associated to the cluster, this list must be empty.

**Required** – No

dataviewName

**Description** – The name of the dataview.

**Type** – String

**Pattern** – `^[a-zA-Z0-9][a-zA-Z0-9-_]*[a-zA-Z0-9]$`

**Length** – 3-63

**Required** – No

Example

You can use this function to specify the changeset that you want to update, as following:

```
.aws.db["example-db";"example-changeset-id"; .aws.cache["CACHE_1000";"/"];""]
```

Alternatively, if the cluster is attached to a dataview, you can use this function to update the cluster to the latest version of the dataview with the specified `dataviewName`, as following:

```
.aws.db["example-db";""; .aws.cache["CACHE_1000";"/"];"example-dataview-name"]
```

Result

The output of this function is used as input for `.aws.sdbs` function.

**Function**: `.aws.cache[cacheType;dbPaths]`

This is a helper function to construct arguments for `.aws.db`.

Parameters

cacheType

**Description** – The type of disk cache. This parameter is used to map the database path to cache storage.

**Type** – String

**Length** – 8-10

**Required** – Yes

dbPaths

**Description** – The portions of database that will be loaded into the cache for access.

**Type** – Array of strings

**Pattern** – `^\/([^\/]+\/){0,2}[^\/]*$`

**Length** – 1-1025

**Required** – Yes

Example

The following two examples show how you can send different requests by using this function.

```
.aws.cache["CACHE_1000";"/"]
```

```
.aws.cache["CACHE_1000";("path1";"path2")]
```

Result

The output of this function is used as input for `.aws.db` function.

**Function**: `.aws.sdbs[deploymentStrategy]`

This is a helper function to construct arguments for `.aws.update_kx_cluster_databases`.

Parameters

deploymentStrategy

**Description** – The type of deployment that you want on a cluster. The following types are available.

- `ROLLING` – This options loads the updated database by stopping the exiting q process and starting a new q process with updated configuration.
- `NO_RESTART` – This option loads the updated database on the running q process without stopping it. This option is quicker as it reduces the turn around time to update a kdb database changeset configuration on a cluster.

**Type** – String

**Required** – Yes

Result

The output of this function is used as input for `.aws.update_kx_cluster_databases` function.

**Pub/Sub APIs**

**Function**: `.aws.sub[table;sym_list]`

Initializes the publish and subscribe function.

Parameters

`table`

**Description** – The symbol of the table that you want to subscribe to. The symbol ` subscribes to all tables.

**Type** – Symbol

**Required** – Yes

`sym_list`

**Description** – The list of symbols to filter published records. Defaults to ` if no filter is applied.

**Type** – Symbol list

**Required** – Yes

Result

Returns table schema or table schemas list.

**Example 1: Subscribes to `tab table and filtering `AAPL`MSFT.**

```
target_instance_connection_handle ".aws.sub[`tab;`AAPL`MSFT]"
```

**Result**

```
`tab
+`sym`sales`prices!(`g#`symbol$();`long$();`long$())
```

**Example 2: Subscribes to all tables with no filtering.**

```
target_instance_connection_handle ".aws.sub[`;`]"
```

**Result**

```
`tab  +`sym`sales`prices!(`g#`symbol$();`long$();`long$())
`tab1 +`sym`sales`prices!(`g#`symbol$();`long$();`long$())
`tab2 +`sym`sales`prices!(`g#`symbol$();`long$();`long$())
`tab3 +`sym`sales`prices!(`g#`symbol$();`long$();`long$())
```

**Function**: `.aws.pub[table;table_records]`

Publishes table records to table subscribers by calling upd`[table;table_records]` within the subscriber connection handle.

Parameters

`table`

**Description** – Publishes the records to the table subscribers.

**Type** – Symbol

**Required** – Yes

`table_records`

**Description** – The table records that you want to publish.

**Type** – Table

**Required** – Yes

Example

This example publishes `` `tab `` table and values to the subscribers.

```
.aws.pub[`tab;value `tab]
```

Result

This function does not return any value.

**Database maintenance APIs**

**Function**: `.aws.commit_kx_database[database_name]`

Commits the database changes after performing database maintenance.

Parameters

`database_name`

**Description** – The name of the database where you performed database maintenance operations and whose changes you want to commit.

**Type** – String

**Required** – Yes

Example

```
.aws.commit_kx_database["welcomedb"]
```

Result

Returns the `id` and `status` of the changeset that the API creates.

# Logging and monitoring

Amazon FinSpace provides a variety of Amazon CloudWatch metrics that you can monitor to determine the health and performance of your FinSpace resources and instances. You can view these metrics using various tools, including the FinSpace console, the AWS CLI, the Amazon CloudWatch console.

## Metrics and dimensions

The `AWS/FinSpace` namespace includes the following metrics.

| Metric | Description |
|---|---|
| `CPUUtilization` | The average CPU utilization across all the nodes in a cluster. |
| `MemoryUtilization` | The average memory utilization across all the nodes in a cluster. |
| `DatabaseCacheDataReadBytes` | The operations for reading database cache. |
| `DatabaseCacheDataWriteBytes` | The operations for writing to database cache. |
| `DatabaseCacheDataReadOperations` | The operations for reading database cache. |
| `DatabaseCacheDataWriteOperations` | The operations for writing to database cache. |
| `DatabaseCacheFreeDataStorageCapacity` | Database cache free storage. |
| `LocalStorageVolumeReadBytes` | The volume for read operation in local storage. |

| Metric | Description |
| --- | --- |
| LocalStorageVolumeWriteBytes | The volume for read operation in local storage. |
| LocalStorageVolumeReadOps | The operations for reading local storage volume. |
| LocalStorageVolumeWriteOps | The operations for writing to local storage volume. |
| MemoryHeapBytes | The memory available in the heap in bytes for the component container. |
| MemoryHeapPeakBytes | The maximum heap size so far in bytes for the component container. |
| MemoryHeapLimitBytes | The memory limit heap in bytes for the component container as set by -w. |
| MemoryMappedBytes | The mapped memory in bytes for the component container. |
| MemoryPhysicalBytes | The physical memory available in bytes for the component container. |
| KdbSymsTotal | The number of symbols for the component container. |
| KdbSymsMemoryBytes | The memory use of symbols in bytes for the component container. |
| KdbHandlesTotal | The number of active connection handles. |
| KdbIpcOpenedTotal | The total number of IPC sockets that have been opened to the component container. |

| Metric | Description |
| --- | --- |
| KdbIpcClosedTotal | The total number of open handles (IPC and WebSocket) for the component container. |
| KdbWsOpenedTotal | The total number of WebSocket connections opened to the component container. |
| KdbWsClosedTotal | The total number of WebSocket connections closed to the component container. |
| KdbSyncTotal | The sync requests made to the component container. |
| KdbHttpGetTotal | The HTTP GET requests made to the component container. |
| KdbHttpPostTotal | The HTTP POST requests made to the component container. |
| KdbWsTotal | The WebSocket messages received by the component container. |
| KdbTsTotal | The timer calls made within the component container. |
| KdbSyncErrTotal | The number of errors returned in sync requests. |
| KdbAsyncErrTotal | The number of errors returned in async requests. |
| KdbHttpGetErrTotal | The number of errors returned in http GET requests. |

| Metric | Description |
| --- | --- |
| KdbHttpPostErrTotal | The number of errors returned in http POST requests. |
| KdbWsErrTotal | The number of errors returned in WebSocket calls. |
| KdbTsErrTotal | The number of errors returned in timer calls. |
| KdbSyncSeconds | The count and time taken by the sync requests. |
| KdbAsyncSeconds | The count and time taken by the async requests. |

The AWS/Usage namespace includes the following metrics.

| Metric | Description |
| --- | --- |
| KxNodeSglargePerEnvironment | The maximum number of kx.sg.large Managed kdb scaling group nodes per environment. |
| KxNodeSgxlargePerEnvironment | The maximum number of kx.sg.xlarge Managed kdb scaling group nodes per environment. |
| KxNodeSg2xlargePerEnvironment | The maximum number of kx.sg.2xlarge Managed kdb scaling group nodes per environment. |
| KxNodeSg4xlargePerEnvironment | The maximum number of kx.sg.4xlarge Managed kdb scaling group nodes per environment. |

| Metric | Description |
| --- | --- |
| KxNodeSg8xlargePerEnvironment | The maximum number of kx.sg.8xlarge Managed kdb scaling group nodes per environment. |
| KxNodeSg16xlargePerEnvironment | The maximum number of kx.sg.16xlarge Managed kdb scaling group nodes per environment. |
| KxNodeSg32xlargePerEnvironment | The maximum number of kx.sg.32xlarge Managed kdb scaling group nodes per environment. |
| KxNodeSgOne16xlargePerEnvironment | The maximum number of kx.sg1.16xlarge Managed kdb scaling group nodes per environment. |
| KxNodeSgOne24xlargePerEnvironment | The maximum number of kx.sg1.24xlarge Managed kdb scaling group nodes per environment. |
| KdbFileStorageVolumesPerEnvironment | The maximum number of Managed kdb volumes per environment. |
| ReadMountsPerKdbVolume | The maximum number of read mounts per Managed kdb volume per environment. |
| WriteMountsPerKdbVolume | The maximum number of write mounts per Managed kdb volume per environment. |
| ManagedKdbVolumeStorage | The maximum amount of storage for Managed kdb volumes per environment. |
| KdbScalingGroupsPerEnvironment | The maximum number of Managed kdb scaling groups per environment. |

| Metric | Description |
| --- | --- |
| KdbDataviewsPerEnvironment | The maximum number of Managed kdb dataviews per environment. |
| ConcurrentKdbDataviewsVersionsPerEnvironment | The maximum number of concurrent Managed kdb dataview version processing. |
| TotalKdbEnvironments | The maximum number of environments per AWS account. |
| ManagedKdbClusterUsers | The maximum number of cluster users per environment. |
| ManagedKdbClusters | The maximum number of clusters allowed per environment. |
| ManagedKdbDatabases | The maximum number of databases allowed per environment. |
| ManagedKdbConcurrentChangesetIngestions | The maximum number of concurrent changeset ingestions allowed per environment. |
| ManagedKdbDatabaseClusterCacheSize | The maximum amount of database cluster cache allowed per environment. |
| ManagedKdbSavedownStorage | The maximum amount of savedown storage allowed per environment. |
| KxSlargeNodes | The maximum number of `kx.s.large` nodes allowed per environment. |
| KxSxlargeNodes | The maximum number of `kx.s.xlarge` nodes allowed per environment. |

| Metric | Description |
| --- | --- |
| KxS2xlargeNodes | The maximum number of kx.s.2xlarge nodes allowed per environment. |
| KxS4xlargeNodes | The maximum number of kx.s.4xlarge nodes allowed per environment. |
| KxS8xlargeNodes | The maximum number of kx.s.8xlarge nodes allowed per environment. |
| KxS16xlargeNodes | The maximum number of kx.s.16xlarge nodes allowed per environment. |
| KxS32xlargeNodes | The maximum number of kx.s.32xlarge nodes allowed per environment. |
| ManagedKdbSingleAzClusters | The maximum number of Single-AZ clusters per environment. |
| ManagedKdbMultiAzClusters | The maximum number of Multi-AZ clusters per environment. |

**Note**

Amazon CloudWatch aggregates these metrics at one-minute intervals.

## Monitoring Managed kdb cluster metrics

You can monitor the health and activity of Managed kdb Insights clusters by using CloudWatch. When you interact with Managed kdb clusters, the following metrics are sent to CloudWatch in the

AWS account that you used to create the Managed kdb environment. You can use the following procedures to view the metrics for Managed kdb clusters.

**To view metrics using the CloudWatch console**

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Under the **All metrics** tab, choose **FinSpace** and then choose the **KxClusterId**.

## Monitoring Managed kdb scaling groups metrics

You can monitor the health and activity of Managed kdb scaling groups by using CloudWatch. When you interact with Managed kdb scaling groups, the following metrics are sent to CloudWatch in the AWS account that you used to create the Managed kdb environment. You can use the following procedures to view the metrics for Managed kdb scaling groups.

**To view metrics using the CloudWatch console**

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Under the **All metrics** tab, choose **FinSpace** and then choose the **KxClusterId**.

FinSpace supports the following metrics for Managed Kdb scaling groups.

| Metric | Description |
| --- | --- |
| `ScalingGroupCpuUtilization` | The total CPU utilization of the Managed kdb scaling group hosts across all clusters. |

| Metric | Description |
| --- | --- |
| ScalingGroupMemoryUtilization | The total memory utilization of the Managed kdb scaling group hosts across all clusters. |

## Monitoring Managed kdb volume metrics

You can monitor the health and activity of Managed kdb volumes by using CloudWatch. When you interact with Managed kdb volumes, the following metrics are sent to CloudWatch in the AWS account that you used to create the Managed kdb environment. You can use the following procedures to view the metrics for Managed kdb volumes.

**To view metrics using the CloudWatch console**

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Under the **All metrics** tab, choose **FinSpace** and then choose the **KxVolumeName**.

FinSpace supports the following metrics for Managed Kdb volumes.

| Metric | Description |
| --- | --- |
| KdbVolumeDataReadBytes | The data in bytes read from managed kdb volumes. |
| KdbVolumeDataWriteBytes | The data in bytes written from managed kdb volumes. |
| KdbVolumeDataReadOperations | The operations for reading Managed kdb volumes. |
| KdbVolumeDataWriteOperations | The operations for writing to Managed kdb volumes. |

| Metric | Description |
| --- | --- |
| KdbVolumeFreeDataStorageCapacity | The Managed kdb volumes free storage. |

## Logging

The KDB application logs from Managed kdb clusters are captured in Amazon CloudWatch Logs. There is a separate CloudWatch Log Group created for each cluster. You can access cluster logs for FinSpace in the following CloudWatch log group.

`/aws/vendedlogs/finspace/<Managed kdb Environment ID>/<Managed kdb Cluster Name>`

You can view data directly in CloudWatch using CloudWatch reporting or CloudWatch Insights. You can also extract the data from CloudWatch into other logging tools like Splunk or Datadog. Using the **Logs**, you can view `stdout` and `stderr` logs the for KDB processes. From this tab, you can navigate to CloudWatch Logs Insights to run custom queries on the logs.

**To view and query logs for Managed kdb clusters**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, under **Managed kdb Insights**, choose **Kdb environments**.
3. From the kdb environments table, choose the name of the environment.
4. On the environment details page, choose the **Clusters** tab. The table under this tab displays a list of clusters.
5. Choose a cluster name to view its logs. The cluster details page opens.
6. Choose the **Logs** tab. The logs are displayed in a list.
7. Choose the **@logstream** links to open the CloudWatch Logs Insights where you can run custom queries on the logs.

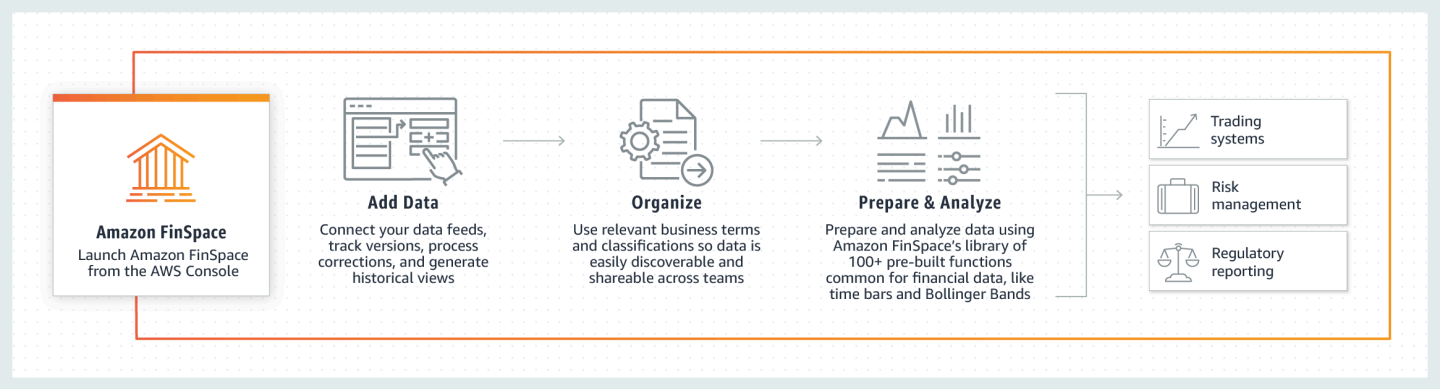
# Dataset browser

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace provides the Dataset browser that you can use to collect data and catalog it by relevant business concepts such as asset class, risk classification, or geographic region; which makes it easy to discover and share across your organization.

## How it works




### To use FinSpace

1. Launch FinSpace from your Amazon Web Services (AWS) console, and configure how data will be organized in the catalog for easy searching.
2. Add data that will be needed for analytics.
3. Organize and describe the data so that it can be searched from the catalog.
4. Prepare data by creating historical or current data views partitioned to optimize performance.
5. Analyze data using integrated Jupyter notebooks, managed Spark clusters, or kdb Insights for data processing at scale.

# Getting started with dataset browser

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Use the topics in this section to create your first Amazon FinSpace environment. A FinSpace environment is your fully managed instance of FinSpace.

**Topics**

- Setting up an Amazon FinSpace environment
- Signing in to the Amazon FinSpace web application
- Using the Amazon FinSpace homepage
- Search and browse data in Amazon FinSpace

## Setting up an Amazon FinSpace environment

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

An Amazon FinSpace environment is created from an AWS account. In this section, you sign up for an AWS account, create an administrator access, and create a FinSpace environment.

**Topics**

- Sign up for Amazon Web Services

- Create an Amazon FinSpace environment
- Sample data bundles

## Sign up for Amazon Web Services

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

When you sign up for AWS, your account is automatically signed up for all services in AWS, including Amazon FinSpace. You are charged only for the services that you use.

If you already have an AWS account, skip to the next step.

### Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

### To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

### Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

#### Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

#### Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

#### Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

### Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.

2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

### (Optional) Attach managed policies for creating FinSpace environment

To create a FinSpace environment, the user performing the actions must have IAM permissions for `AdministratorAccess` or must have the FinSpace managed policy attached to their role. This step is optional if the user has `AdministratorAccess` permissions. Create and attach FinSpace managed policies to the account you used to create the FinSpace environment. These policies grant permissions to create the FinSpace environment and superusers in an AWS account.

1. Create a managed policy on the JSON tab for FinSpace. For more information, see Creating policies on the JSON tab.
2. Use the following managed policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "finspace:*"
            ],
            "Resource": "*"
        }
    ]
}
```

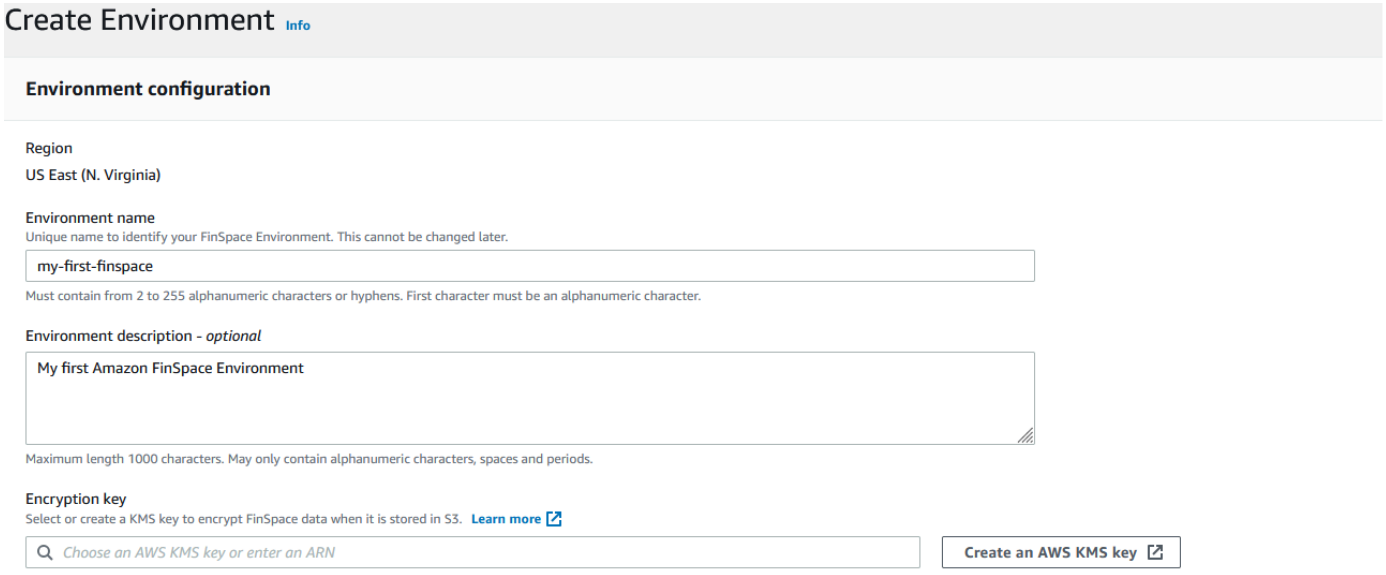
# Create an Amazon FinSpace environment

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

An Amazon FinSpace environment is created from an AWS account. To create a FinSpace environment, the user performing the actions must have IAM permissions for `AdministratorAccess` or the FinSpace managed policy attached to their role.

### To create a FinSpace environment

1. Sign in to your AWS account and open FinSpace from the AWS Management Console. It is located under Analytics, and you can find it by searching for *FinSpace*. Your AWS account number is displayed for verification purposes.
2. Choose **Create Environment**.



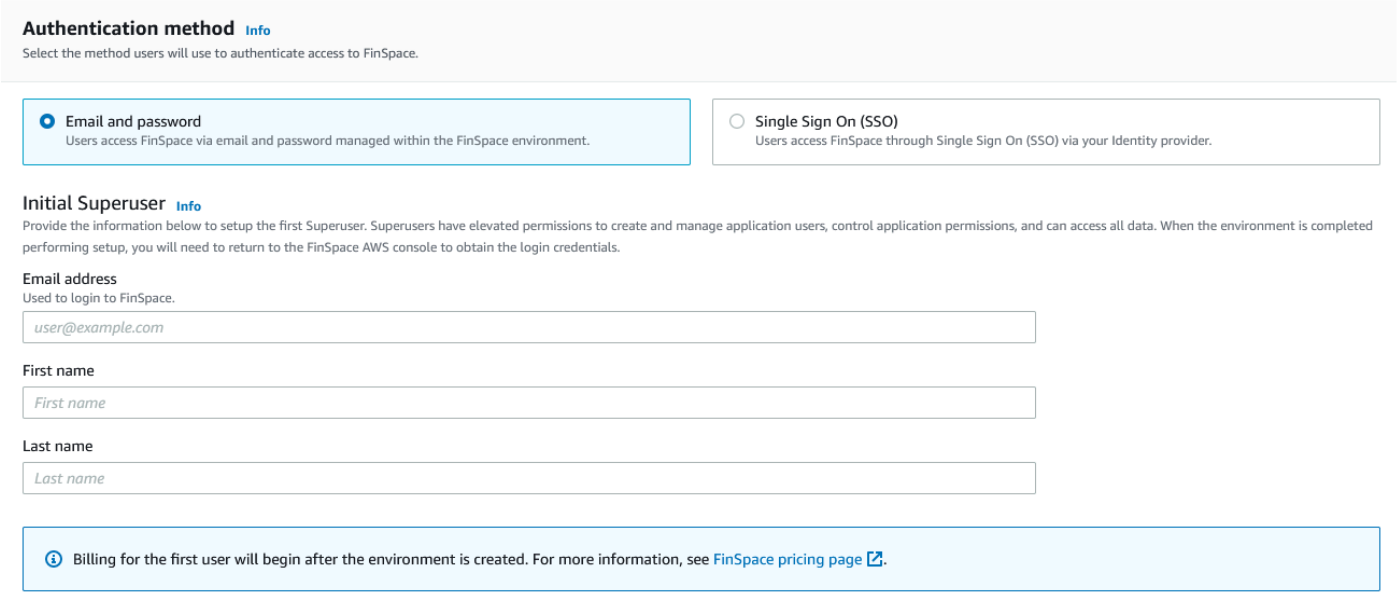

3. Enter a name for your FinSpace environment under **Environment name**.
4. (Optional) Add **Environment description**.

5. Select a symmetric encryption KMS key to encrypt data in your FinSpace environment. If a KMS key is not available in the region where you want to create your FinSpace environment, create a new key.

   For more information, see Creating keys in the *AWS Key Management Service Developer Guide*

6. Select an authentication method for the environment from the following options:

> **Warning**
>
> Selected authentication method cannot be changed once an environment is created.




   a. **Email and password**: You must specify an initial superuser. A superuser has elevated permissions to create and manage application users, control application permissions and access all data. When the environment is completed performing setup, you will need to return to the FinSpace AWS console to obtain the sign in credentials from the environment details page. Enter the following information for the superuser:

      i. Enter the **Email address**.

      ii. Enter **First name**.

      iii. Enter **Last Name**.

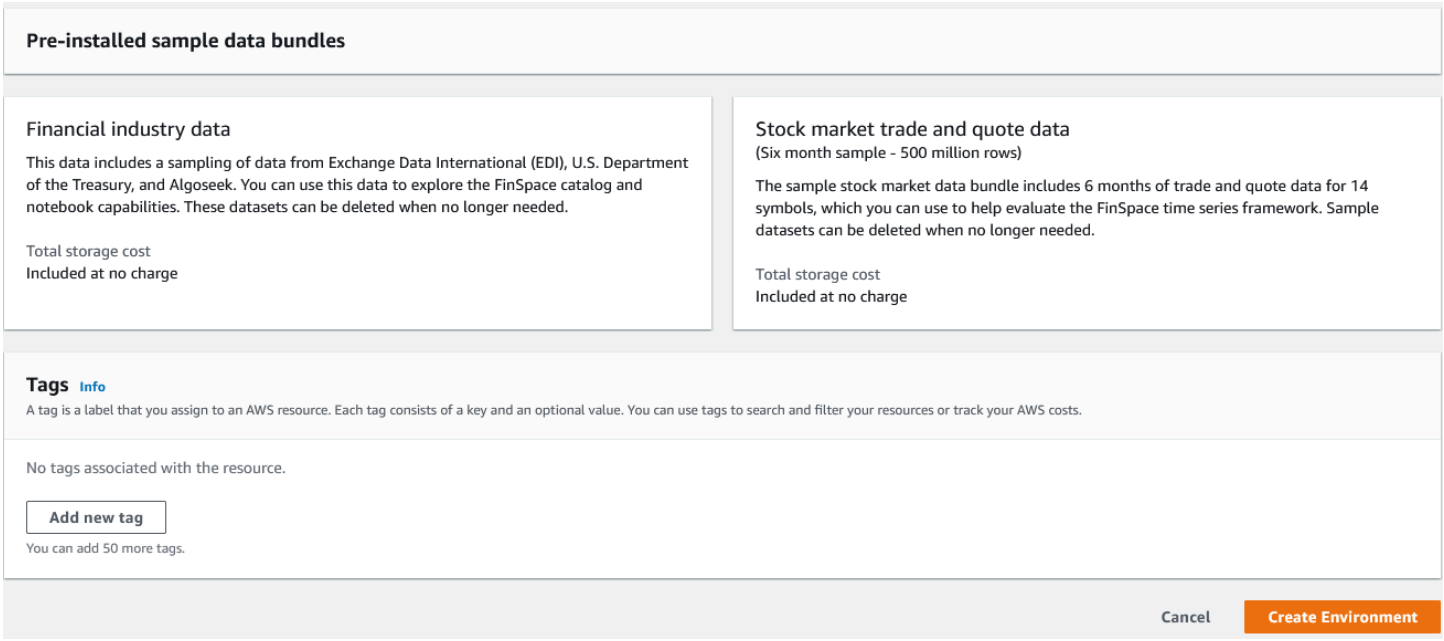
   b. **Single Sign On**:

i. Enter the name of your SAML 2.0 Identity Provider (IdP) which will be used for authentication.

ii. You can choose to either upload SAML metadata document or enter the SAML metadata document URL issued by your IdP. Learn more about SAML 2.0 based SSO support in FinSpace.

iii. Provide the attribute definition from your SAML 2.0 compliant identity provider (IdP) for the email field. Refer to the documentation of your IdP to determine the correct format for the attribute. An example for email attribute is `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`.

7. Choose **Create Environment**. The environment creation process has now begun and it will take 50-60 minutes to finish in the background. You can return to other activities while the environment is being created.




After the environment is created, a domain URL will be generated which is the sign-in url for your FinSpace web application.

> **Note**
>
> Review Inter-network traffic privacy in Amazon FinSpace Dataset browser to ensure that your FinSpace web application is accessible to users.

### Setup additional superusers

After your Amazon FinSpace environment is created, you can create additional superusers and configure permission groups from within the FinSpace web application. A superuser has all permissions to take all actions in FinSpace. The first superuser is created when the environment is created in the AWS console page. After the superuser is created, the superuser uses the credentials to login to the FinSpace web application for the first time.

**To create a superuser**

1. Sign in to your AWS account in which the FinSpace environment was created and open FinSpace from the AWS management console. It is located under Analytics, and you can find it by searching for FinSpace. Your AWS account number is displayed for verification purposes.
2. Select the FinSpace environment for which a superuser will be created.
3. In the section, superusers, choose **Add superuser.**
4. Enter the **Email address**.
5. Enter **First name**.
6. Enter **Last name**.
7. Choose **Next**.
8. Review the superuser details.
9. Choose **Create and view credentials**. Note that if you have created an environment with SSO, you will not receive a temporary password as you will be authenticated with your IdP.

The credentials of superusers, who have yet to sign in, are listed in a banner at the top of the environment details page.

Share the credentials with the person designated as the superuser. The credentials are necessary to sign in to your FinSpace web application. The **Domain** is the sign-in url for your FinSpace web application.

### AWS tags

You can optionally assign tags to an Amazon FinSpace environment. A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define. If you're using AWS Identity and Access Management, you can control which users in your AWS account have permission to create, edit, or delete tags.

**To add a new tag in your FinSpace environment**

1. Sign in to your AWS account and open FinSpace from the AWS Management Console. It is located under Analytics, and you can find it by searching for *FinSpace*. Your AWS account number is displayed for verification purposes.
2. Select the FinSpace environment to manage and add tags.
3. Under the **Tags** section, choose **Manage Tags**.
4. To add a new tag, choose **Add new tag**. Add tag details.
5. Choose **Save changes**.

**To delete an existing tag in your FinSpace environment**

1. Sign in to your AWS account and open FinSpace from the AWS Management Console. It is located under Analytics, and you can find it by searching for FinSpace. Your AWS account number is displayed for verification purposes.
2. Select the FinSpace environment to manage and add tags.
3. Under the **Tags** section, choose **Manage Tags**.
4. Choose **Remove** for the tag you want to remove.
5. Choose **Save changes**.

## Sample data bundles

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

All environments have a Capital Markets Sample data bundle installed so you can browse, search and analyze this data to explore FinSpace.

The Capital Markets Sample data bundle includes sample datasets that contain trades and quotes data, example categories and controlled vocabularies. The sample datasets can also be used with the provided example notebooks.

# Signing in to the Amazon FinSpace web application

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You sign in to the Amazon FinSpace web application by using the credentials and domain sign-in url provided by your FinSpace superuser.

**To sign in to FinSpace**

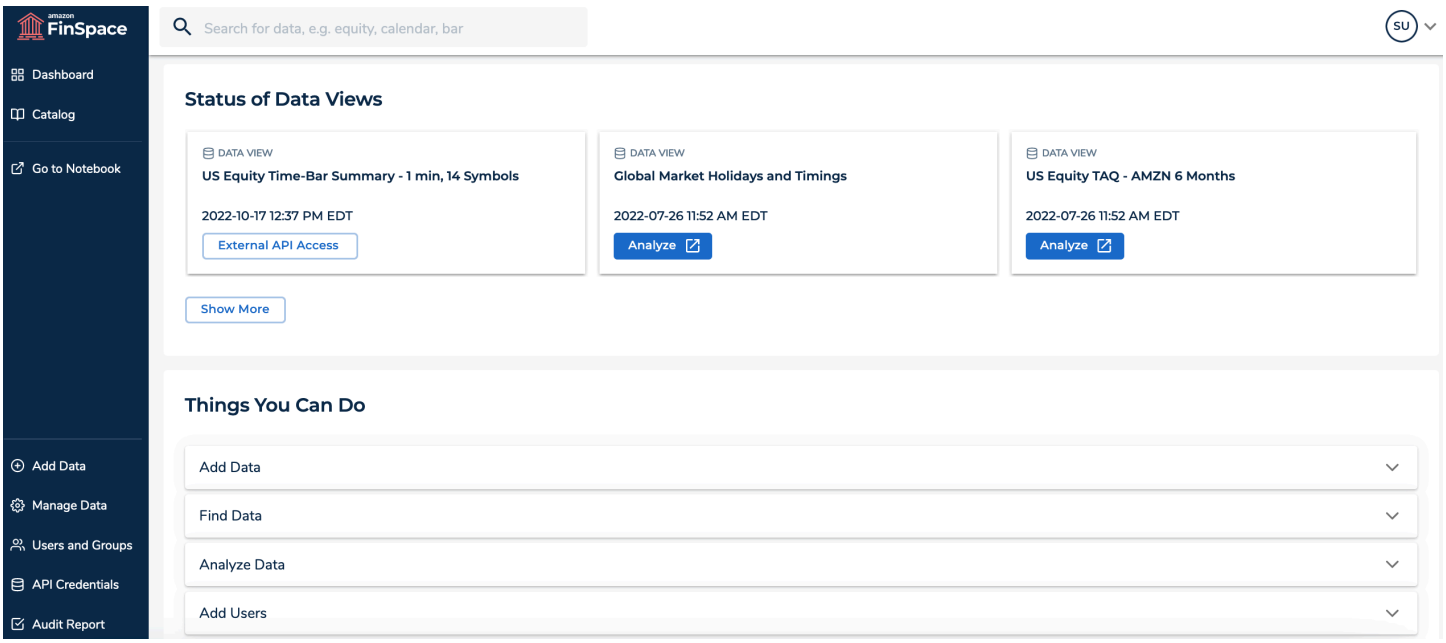
1. Visit the sign-in url on your browser. If you are redirected to sign in through Single Sign On with your Identity provider, enter your corporate credentials to sign in. If the credentials do not work, contact your FinSpace superuser.
2. If you are directed to a page that requires you to enter email and password, follow steps below.
3. Enter the **Email** that was provided with credentials.
4. If you are not signing in for the first time, skip to step 6. If you are signing in for the first time, proceed to next step.
5. Enter the temporary password provided by your superuser.
6. Enter your password in **Password**. If you aren't sure, ask the superuser. If you create a new password, enter your password again to confirm it. Passwords are case-sensitive, must be between 8 and 64 characters in length, and must contain at least one character from four of the following categories:

   - Lowercase letters (a-z)
   - Uppercase letters (A-Z)
   - Numbers (0-9)
   - Special characters (`= + - ^ $ * . [ ] { } ( ) ? " ! @ # % & / \ , > < ' : ; | _ ~ ` )`

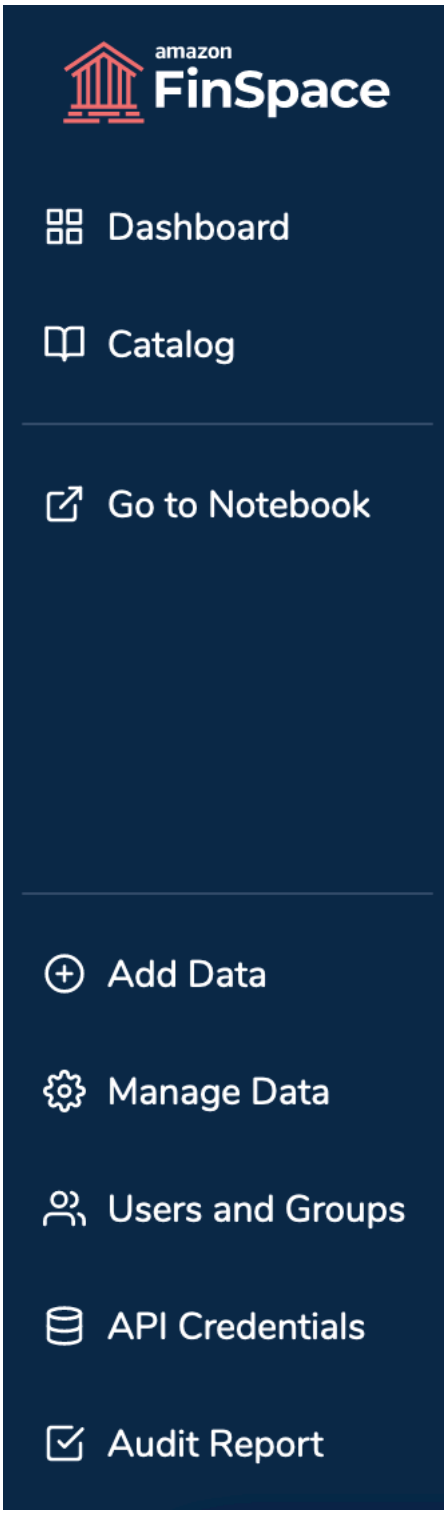
# Using the Amazon FinSpace homepage

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

When you sign in to the Amazon FinSpace web application, you see the FinSpace homepage. For details on how to sign in, see Signing in to the Amazon FinSpace web application. This section walks you through the various parts of the homepage. Note that most features are enabled by permissions and if your user is not a member of a permission group with permissions, such as **Access Notebooks**, you will not see the **Go to Notebook** button at the left side of the homepage. For more information on permissions, see Managing user permissions with permission groups.

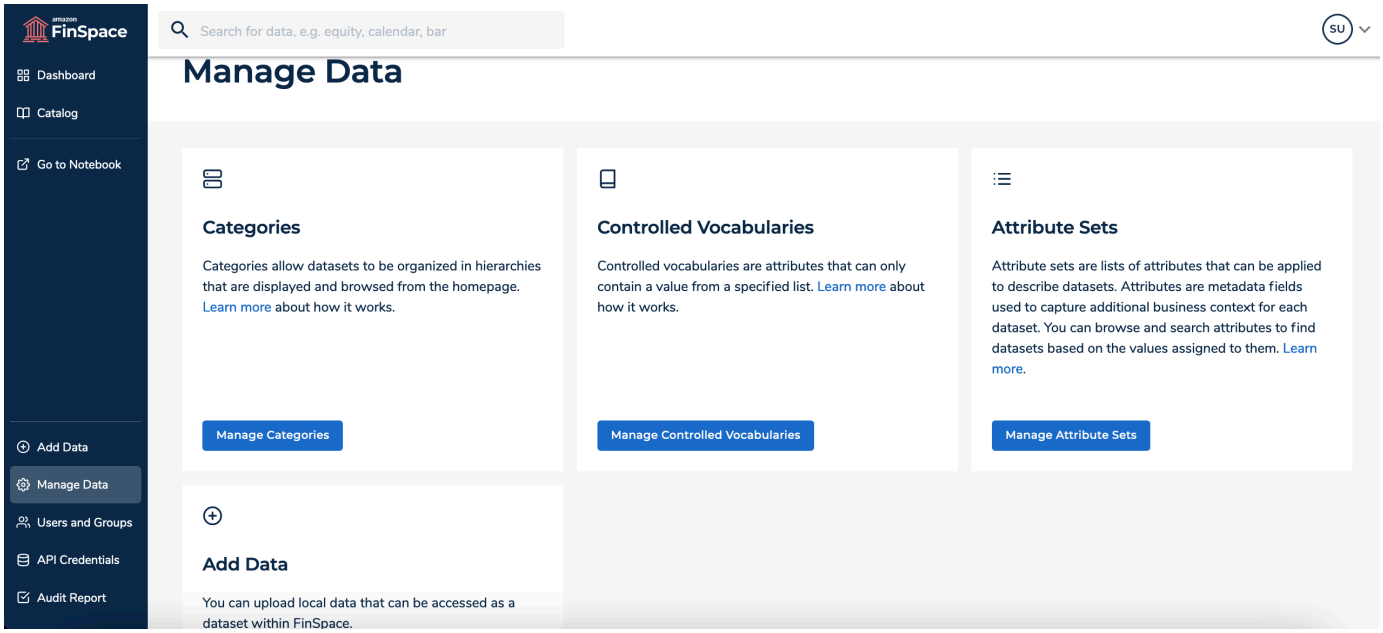


## Left navigation bar




The navigation bar on the left consists of the following controls:

- **Amazon FinSpace icon** – Is located in the top left corner and functions as a home link. Choosing this icon from anywhere in the application returns you to the homepage.
- **Dashboard** – Opens a dashboard view of the homepage.
- **Catalog** – Opens the data browser and the browse results page.
- **Go to Notebook** – Opens a FinSpace notebook in a new tab of your browser. This control is visible only if your user is a member of a permission group with necessary permissions.
- **Administrative controls** – The left navigation consists of the following controls that provide access to the administrative functions in FinSpace. Each menu item will take you to the function for that feature. These functions will be visible on the menu only if your user is a member of a permission group with necessary permissions.
  - **Add Data** – Opens the **Add Data** page, where you can quickly upload a data file and create a new dataset to store the data file. For more information, see Adding and managing data in Amazon FinSpace.
  - **Manage Data** – Opens the **Manage Data** page, where you can configure a business data catalog for browsing datasets by using categories, controlled vocabularies, and attribute sets. You can also add data from this page.




- **Users and Groups** – Opens the **Users and Permission Groups** page, where you can create permission groups and assign users. For more information, see Managing user permissions with permission groups.

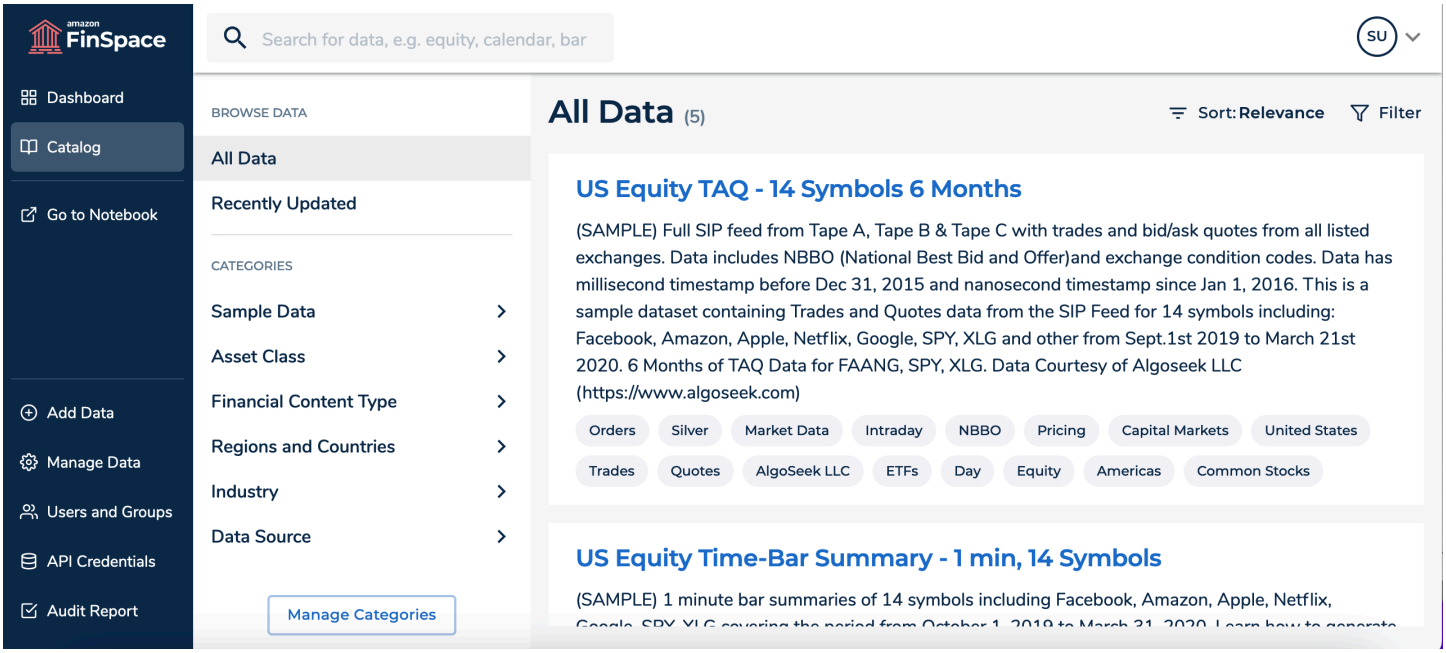
- **API Credentials** – Opens the **API Credentials** page, from where you can get the credentials to access the FinSpace data API operations. These credentials are only valid for 60 minutes. After the credentials expire, you need to choose the refresh icon to generate new credentials.
- **Audit Report** – Opens the **Generate Audit Report** page, from where you can generate audit reports to identify the type of user activity that has occurred within FinSpace for a period of time.

## Top navigation bar


The top navigation bar consists of the following controls:

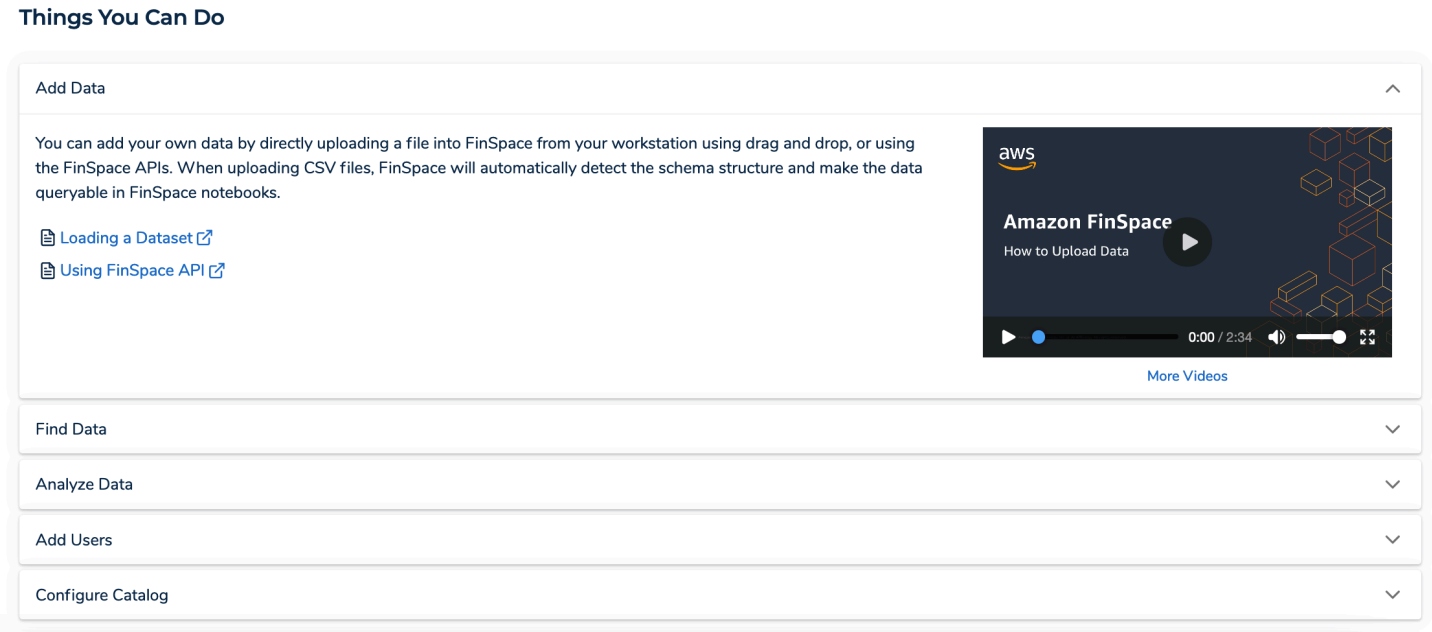
- **Keyword search box** – This search box enables you to enter text to search for datasets in FinSpace.
- **User profile menu** – The user profile menu on the far right of the navigation bar that shows your user initials provides access to your user profile, links to the documentation, tutorial videos about using FinSpace, and the ability to log out of FinSpace. If you select your name you can see what permission groups you are a member of.

## Catalog



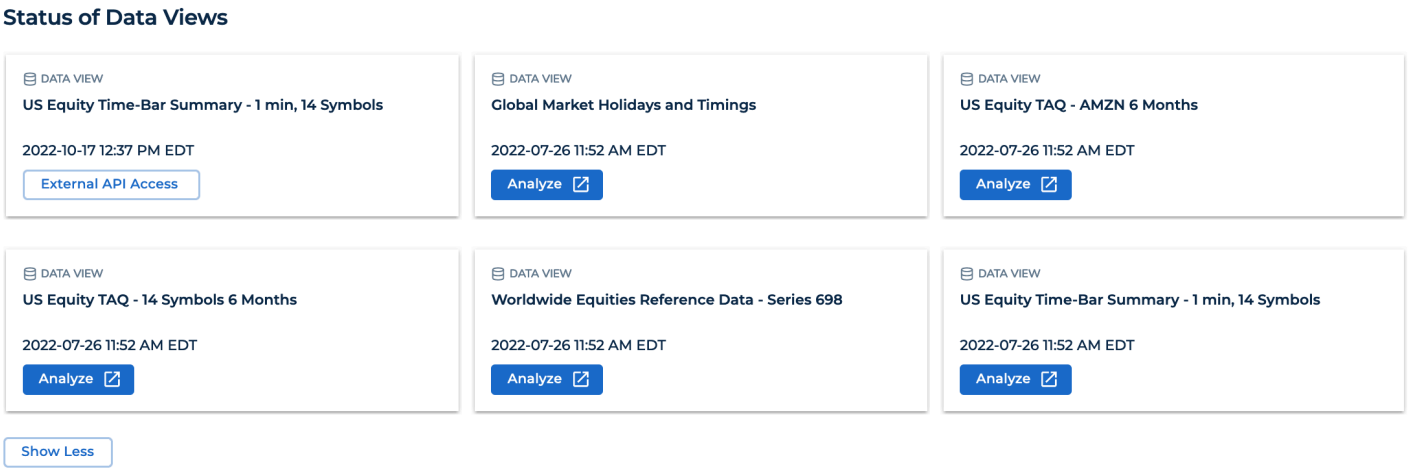
Choosing this button takes you to the browse results page that contains the data browser, where you can browse datasets with categories that you can configure yourself. Selecting any of these nodes will take you to a results page that will find you all the datasets in FinSpace that are associated with that selected category.

## Action cards




In the bottom section of the homepage you will find action cards titled **Add Data**, **Find Data**, **Analyze Data**, **Add users**, and **Configure Catalog**. Each card provides guidance to help you get started with FinSpace.

## Status of data views



This section shows your most recently created data views of datasets including the status of processing when you create a new view. You can also display views with partitions and sorting by choosing schema columns at the time of creating a data view. You can choose the dataset name to go to the dataset details page. The **Analyze** button at the bottom of the card allows you to access a notebook with a sample code to access the view.

If you select to access the data view externally using the FinSpace API while creating the data view, you will see the **External API Access** button at the bottom of the card. Choose this button to access the data view using your FinSpace API credentials.

## Search and browse data in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace provides you the ability to search for data using key words or you can browse for data using the data browser which displays the categories defined in your environment.

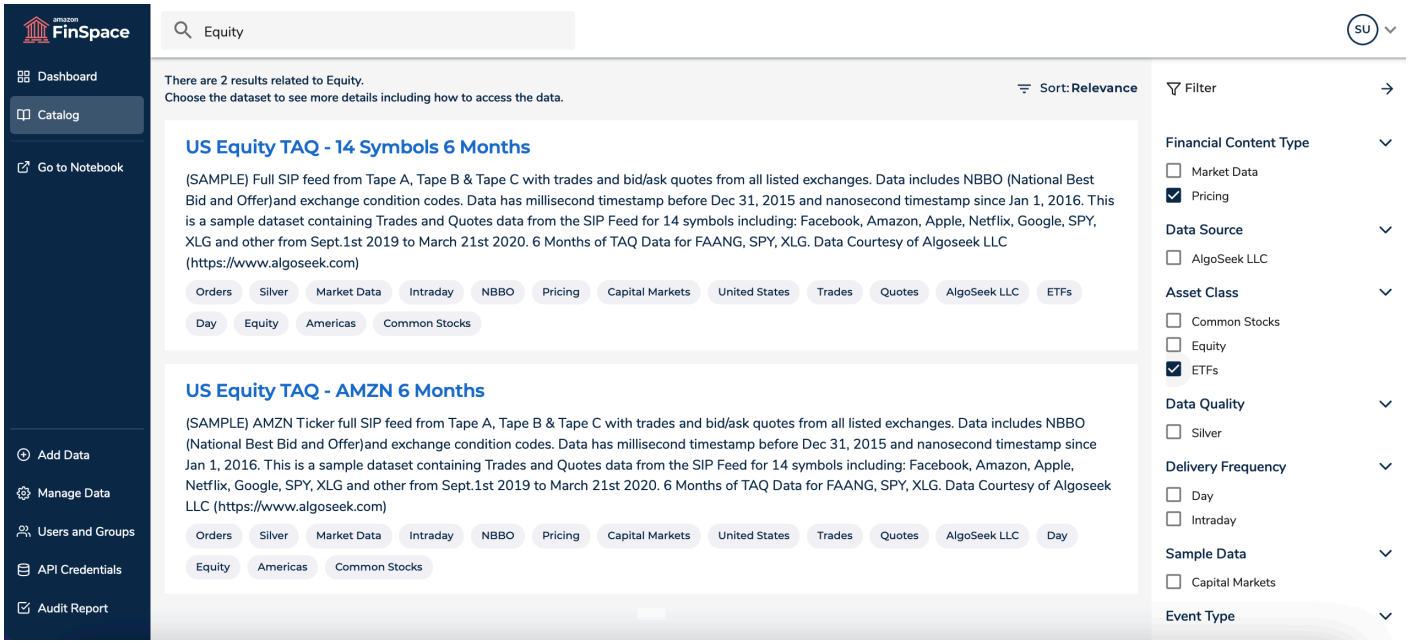
### Search bar

#### To search for datasets in FinSpace web application

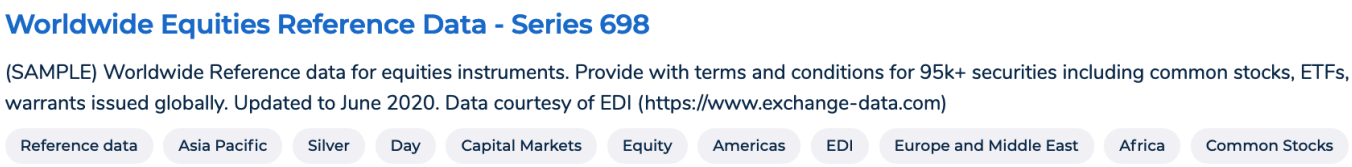
1. From the homepage, search for data using the search box on the top navigation bar.
2. When you type a keyword, for example `equity`, recent searches will appear on the **Catalog** page. Type * for requesting all datasets. To search for all datasets starting with word `Equity`, type `Equity*`.
3. Use the return key to receive results. If you have datasets matching the keyword, results will be returned on the screen. The results will differ depending on the permissions assigned to a permission group that you are a member of.

   For example, a superuser will see all datasets while an application user will only see a dataset if they are a member of a group with read permission for that dataset.

You can sort results by recently updated datasets, relevance, alphabetical order.



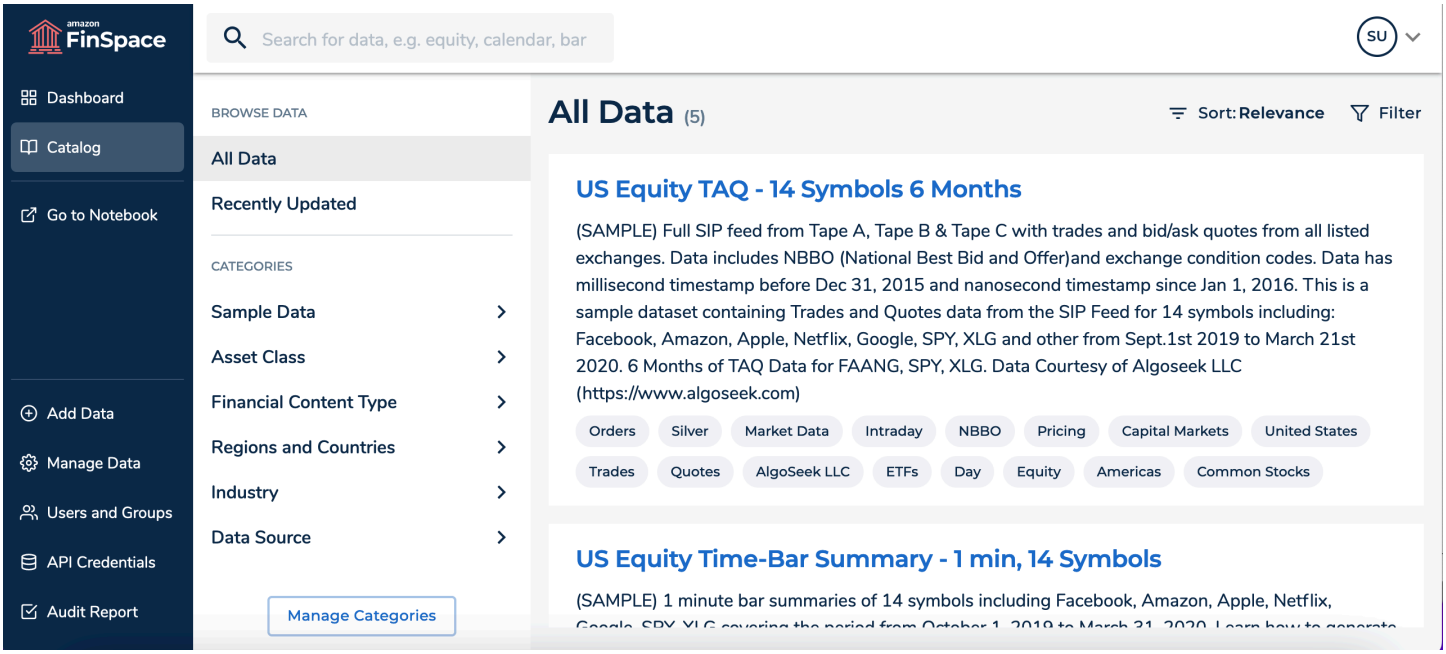

4. The right panel shows search filtering options that are created based on the attribute sets associated to the returned datasets.
5. You can search for related datasets by choosing the dataset attributes tags of the dataset result.




6. To view the details of a dataset in the results, choose the name of the dataset that is displayed in bold.

## Data browser

You can also find datasets by using the data browser on the **Catalog** page. This is set up by your organization for users to easily search for datasets. All users will see the same categories in the data browser. However, when browsing the categories they will only see datasets to which they have appropriate permissions. For example, a superuser will see all datasets, while an application user will only see a dataset if they are a member of a group with read permission for that dataset.




You can search for related datasets by choosing the dataset attributes in the dataset attributes tags of the dataset result.

To view the details of a dataset in the result, choose the name of the dataset that is displayed in bold.

# Understanding datasets in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Datasets are described, organized, and made browsable and searchable in Amazon FinSpace with three constructs:

- **Categories** – Categories allow for cataloging of datasets by commonly used business terms. Categories are hierarchical in nature, allowing for each node of the hierarchy to have a name and a description. The order of the nodes within a level are defined when you define categories. The

**Categories** are displayed in the data browser when you choose **Catalog** on the left navigation bar.

- **Controlled Vocabularies** – Controlled Vocabularies are enumeration lists of attributes to describe datasets.
- **Attribute Sets** – Attribute Sets are lists of attributes that can be applied to datasets. Attributes are metadata fields used to capture additional business context for each dataset. You can then browse and search attributes to find a dataset based on the values assigned to the attributes.

For information on how to configure your business catalog see Tutorial: Configuring a business data catalog in Amazon FinSpace.

**Topics**



## Configuring categories in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

**Categories** – Categories allow for cataloging of datasets by commonly used business terms. Categories are hierarchical in nature, allowing for each node of the hierarchy to have a name and a description. The order of the nodes within a level are defined when you define categories. The **Categories** are displayed in the data browser when you choose **Catalog** on the left navigation bar.

> **Note**
>
> In order to create and manage categories, you must be a superuser or a member of a group with necessary permissions - **Manage Categories and Controlled Vocabularies**.

## Create a category

### To create a new category

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Categories**.
4. Choose **Add New Top Level Category**.
5. Enter a name for the category. For example, `Asset Class`.
6. (Optional) Add a description for the category.
7. Choose **Add Sub-Category** to add one or more sub-categories. An addition of one category is required. You can add as many sub-categories as you like.
8. (Optional) Add a description for the sub-category.
9. Choose **Done** to add the sub-category.
10. Choose **Save**.

## Display category in the data browser

### To display a category in the data browser

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Categories**.
4. Identify the top level category that you want to make visible in the data browser.

5. Uncheck the eye
   (  )
   icon.
6. On the left navigation bar, choose **Catalog** and verify if the category is now visible in the data browser.

## Editing categories

### To edit a category

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Categories**.
4. From the list of categories, select a category to edit.
5. Choose **Edit**.
6. In the **Edit Category** section, make the required changes.
7. Choose **Save**.

## Deleting categories

### To delete a category

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Categories**.
4. From the list of categories, select a category that you want to delete.
5. Choose **Edit**.
6. Choose **Delete**.
7. On the dialog box that appears, choose **Remove** to confirm deletion.

# Configuring controlled vocabularies in Amazon FinSpace

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Controlled Vocabularies are enumeration lists of attributes to describe datasets.

**Note**

In order to create and manage controlled vocabularies, you must be a superuser or a member of a group with necessary permissions - **Manage Categories and Controlled Vocabularies**.

## List all controlled vocabularies

### To list all controlled vocabularies

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Controlled Vocabularies**. All the available controlled vocabularies are listed in a table from where you can edit or duplicate them.

## Create controlled vocabularies

### To create controlled vocabularies

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.

3. On the **Manage Data** page, choose **Manage Controlled Vocabularies**.
4. On the **Controlled Vocabularies** page, choose **Create Controlled Vocabulary**.
5. Enter a name for the controlled vocabulary. For example, `US States`.
6. Choose **Add Field** to add different values for your controlled vocabulary. For example, you can add `Arizona`, `Alaska` etc. as the field names for the controlled vocabulary `US States`.
7. Choose **Save**.

## Edit controlled vocabularies

### To edit controlled vocabularies

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Controlled Vocabularies**.
4. On the **Controlled Vocabularies** page, select a controlled vocabulary to edit.
5. Choose **Edit**.
6. In the **Edit Controlled Vocabulary** section, make the required changes.
7. Choose **Save**.

## Delete controlled vocabularies

### To delete controlled vocabularies

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Controlled Vocabularies**.
4. From the list of controlled vocabularies, select the one that you want to delete.
5. Choose **Edit** and then choose **Delete**.
6. On the dialog box that appears, choose **Remove** to confirm deletion.

# Configuring attribute sets in Amazon FinSpace

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Attribute sets are lists of attributes that can be applied to describe datasets. Attributes are metadata fields used to capture additional business context for each dataset. You can browse and search attributes to find datasets based on the values assigned to them.

**Note**

In order to create attribute sets, you must be a superuser or a member of a group with necessary permissions - **Manage Attribute Sets**.

## List all attribute sets

### To list attribute sets

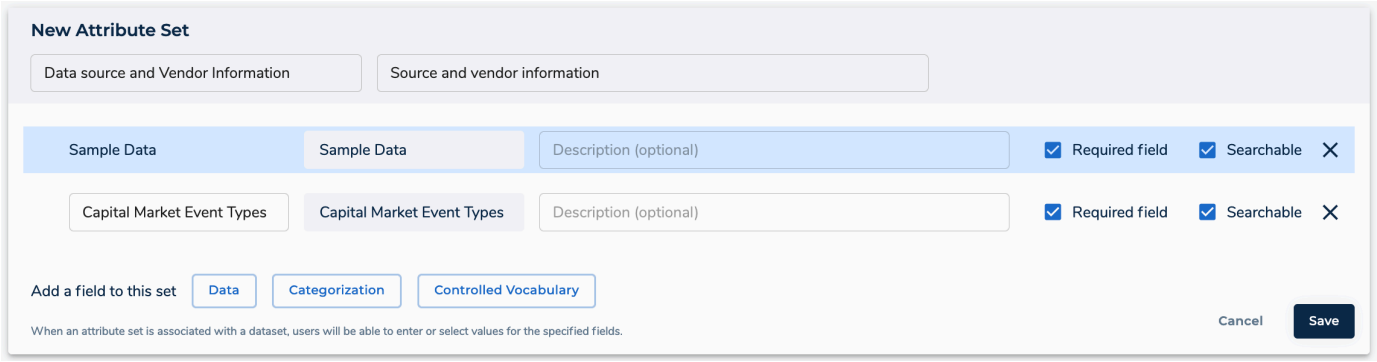
1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Attribute Sets**. All the available attributes are listed in a table. You can also choose the more () icon for options to duplicate, disable, or remove an attribute set.

## Create attribute sets

### To create an attribute set

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Attribute Sets**.
4. On the **Attribute Sets** page, choose **Create Attribute Set**.
5. Enter a name for the attribute set. For example, `Data source and Vendor Information`.
6. (Optional) Add a description for the attribute.
7. Choose the type of fields to the attribute set.

   - **Data** – A field of type Number, String, or Boolean.
   - **Categorization** – A field that is a type of an already defined category. For example, an `Asset class`.
   - **Controlled Vocabulary** – A field that is a type of an already defined controlled vocabulary. For example, `Data Sensitivity Classification`.




8. Choose **Save**.

## Edit attribute sets

### To edit an attribute set

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.

2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Attribute Sets**.
4. On the **Attribute Sets** page, select an attribute set from the list to edit.
5. Choose **Edit This Attribute Set**.
6. Select an existing field to edit and remove it. You cannot change the data type for a field.
7. Choose **Save**.

## Delete an attribute set

### To delete an attribute set

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Attribute Sets**.
4. From the list of attribute sets in the table, select the one that you want to delete.
5. Choose the more () icon and then choose **Remove**.
6. On the dialog box that appears, choose **Remove** to confirm deletion.

## Associate an attribute set with a dataset

### To associate an attribute set to a dataset

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the homepage, search for a dataset using the search box.
3. On the **Catalog** page, choose the dataset name to view the dataset details page.
4. On the **Data Overview** tab, under **Details About This Dataset** section, choose the **Add Attribute Set** button.
5. From the drop down menu, select one or more attribute set to add.

6. Choose **Add Attribute Set**. The selected attribute sets get added to the dataset. You can set values for each of the attribute set you selected and choose **Save** to finish.

> **Note**
>
> Under the **Details About This Dataset** section, you can also choose **Edit** to edit the associated attribute or choose **Remove Attribute Set** to remove it from the dataset.

## Tutorial: Configuring a business data catalog in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The section outlines the procedures to configure a business data catalog for browsing datasets by using categories, controlled vocabularies, and attribute sets. You will take following steps in this tutorial:

- Create categories – `Data Types` and `Data Source`.
- Create controlled vocabulary – `Data Classification`.
- Create attribute set with attributes of type category and controlled vocabulary – `Data` and `Source Information`.
- Create a dataset – `Industrial production total index`.
- Associate attribute set with the newly created Dataset.
- Verify if the dataset is accessible from business data catalog via categories menu.

### Prerequisites

Before you begin, learn about the concepts that are used for configuring a business data catalog by referring to Core concepts and terms.

> **Note**
>
> In order to use this tutorial, you must be a member of a group with the necessary permissions - **Create Datasets**, **Manage Categories and Controlled Vocabularies**, **Manage Attribute Sets**.

## Step 1: Create categories in Amazon FinSpace

Use the following procedures to create the `Data Types` and `Data Source` categories.

### To create the `Data Types` Category

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Categories**.
4. Choose **Add New Top Level Category**.
5. For category name, enter `Data Types`.
6. (Optional) Add a description for the category. For example, you can enter `Type of data`. This will show up as a tool tip when hovering over the menu.
7. Choose **Add Sub-Category** to add one or more sub-categories. You can add as many sub-categories as you like. In this example, for sub-category names enter `Economic Data`, `Commodities Data`, and `Alternative Data`.
8. Choose **Done** to add the sub-category.
9. Choose **Save**.

### To create the `Data Source` category

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Manage Data**.
3. On the **Manage Data** page, choose **Manage Categories**.
4. Choose **Add New Top Level Category**.
5. For category name, enter `Data Source`.

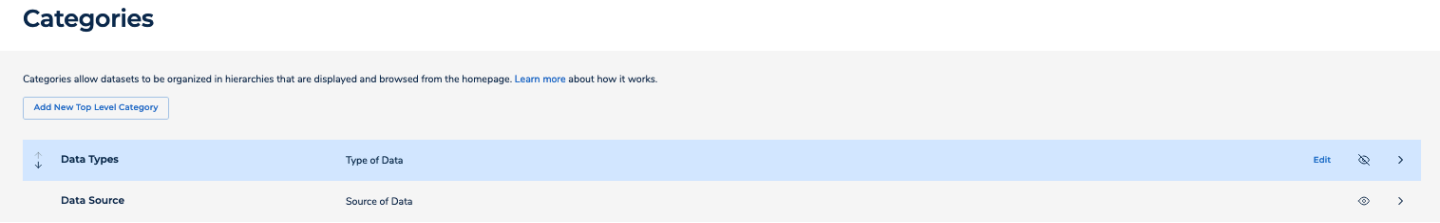
6. (Optional) Add a description for the category. For example, you can enter `Source of data`. This will show up as a tool tip when hovering over the menu.
7. Choose **Add Sub-Category** to add one or more sub-categories. You can add as many sub-categories as you like. In this example, for sub-category names enter `Central Bank`, `Vendor`, and `Exchange`.
8. Choose **Done** to add the sub-category.
9. Choose **Save**.

#### Change visibility of the categories in data browser

On the **Categories** page, uncheck the eye

( )

icon for both categories that you created in above procedures to make them visible in the data browser.




## Step 2: Create controlled vocabulary in Amazon FinSpace

#### To create the `Data Classification` controlled vocabulary

1. On the left navigation bar of the home page, choose **Manage Data**.
2. On the **Manage Data** page, choose **Manage Controlled Vocabularies**.
3. Choose **Create Controlled Vocabulary**.
4. For vocabulary name, enter `Data Classification`.
5. (Optional) Add a description for the vocabulary. For example, you can enter `Data Classification scheme`.
6. Choose **Add Field** to add one or more fields under a vocabulary. You can add as many fields as you like. In this example, for field names enter `Public Data`, `Internal Data`, and `Restricted Data`.
7. Choose **Save**.

## Step 3: Create attribute sets in Amazon FinSpace

**To create the attribute set for `Data and Source Information`**

1. On the left navigation bar of the home page, choose **Manage Data**.
2. On the **Manage Data** page, choose **Manage Attribute Sets**.
3. Choose **Create Attribute Set**.
4. For attribute name, enter `Data and Source Information`.
5. Choose **Categorization** to add a categorization field type.
6. On the **Add Categorization Field** page, do the following:

   a. Choose **Data Types** as the categorization field type.

   b. Choose **Add Field**.

   Repeat these steps again and choose **Data Source** as the as the categorization field type.
7. Choose **Controlled Vocabulary** to add a controlled vocabulary field type.
8. On the **Add Controlled Vocabulary Field** page, do the following:

   a. Choose **Data Classification** as the controlled vocabulary field type.

   b. Choose **Add Field**.
9. Choose **Save**.

## Step 4: Create a dataset in Amazon FinSpace

**To create a dataset**

1. On the left navigation bar of the home page, choose **Add Data**. The source of the data is Federal reserve bank of St.Louis.
2. Drag and drop the Industrial production total index.csv file on the page or choose **Browse Files** to select a new file.
3. On the **Add Data** page, verify if the derived schema is correct.
4. If the derived schema is incorrect, choose **Edit Derived Schema** to edit it.

   For example, in this sample file, the inferred data type for the column **date** is **String**, change it to **Date**.

5. After editing the schema, choose **Save Schema**.
6. Choose an appropriate perimission group that should be associated to the dataset when it gets created. You can add additional permission groups after the dataset creation is complete.
7. Choose **Confirm Schema & Upload File**.

   This action creates a dataset with name `Industrial production total index` and takes you to the Dataset details page.

   > **Note**
   >
   > For small files of up to 100 megabytes, data view creation takes approximately 2 minutes. For larger files of around 1 gigabyte, expect data view creation to take approximately 3-4 minutes. Views with partitioning and sorting schemes may take longer.

   Once the upload of the sample data file is complete, a process is kicked off to create a data view that can be analyzed in a notebook.

## Step 5: Associate an attribute set with a dataset in Amazon FinSpace

**To associate `Data and Source Information` attribute set with `Industrial production total index dataset`**

1. On the homepage, search for `Industrial production total index` dataset in the search box.
2. On the **Catalog** page, choose **Industrial production total index** from the results, to go to the Dataset details page.
3. On the dataset details page for `Industrial production total index`, under **Details About This Dataset**, choose **Add Attribute Set**.
4. On **Add Attribute Set** page, do the following

   a. From the drop down menu, choose `Data and Source Information`.

   b. Choose **Add Attribute Set**.
5. Edit the values for `Data and Source Information` as following:

a. For **Data Types**, enter `Economic Data`.

b. For **Data Source**, enter `Central Bank`.

c. For **Data Classification**, enter `Public Data`.

6. Choose **Save**.

### Step 6: Search the dataset from data browser in Amazon FinSpace

**To search dataset `Industrial production total index` using the data browser**

1. On the left navigation bar of the home page, choose **Catalog**.
2. On the **Catalog** page, under **CATEGORIES** on the data browser, choose the **Data Types** drop down.
3. Choose **Economic Data**. You should see **Industrial production total index** on the right.

   Your business data catalog is now ready. The `Industrial production total index` dataset is now discoverable from the data browser.

## Loading and analyzing data in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Before you proceed with this section, we recommend that you begin by reading Adding and managing data in Amazon FinSpace.

**Use the following procedure to**

- Add sample data, create dataset, and data view using a CSV file. You can upload a CSV file of up to 2 GB directly from the FinSpace web application to add data.
- Analyze the data view in Amazon FinSpace notebook.

> **Note**
>
> In order to perform these steps, you must be a member of a permission group with the necessary permissions - **Create Datasets**, **Manage Clusters**, **Access Notebooks**.

## Add data, create dataset, and data view

### To add data

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Add Data**.
3. On the next page, drag and drop the Industrial production total index.csv file on the page or choose **Browse Files** to select a new file.
4. On the **Add Data** page, verify if the derived schema is correct.
5. If the derived schema is incorrect, choose **Edit Derived Schema** to edit it.

   For example, in this sample file, the inferred data type for the column **date** is **String**, change it to **Date**.
6. After editing the schema, choose **Save Schema**.
7. Choose an appropriate permission group that should be associated to the dataset when it gets created. You can add additional permission groups after the dataset creation is complete.
8. Choose **Confirm Schema & Upload File**.

   This action creates a dataset with name **Industrial production total index** and takes you to the Dataset details page.

   > **Note**
   >
   > For small files of up to 100 megabytes, data view creation takes approximately 2 minutes. For larger files of around 1 gigabyte, expect data view creation to take approximately 3-4 minutes. Views with partitioning and sorting schemes may take longer.

Once the upload of the sample data file is complete, a process is kicked off to create a data view that can be analyzed in a notebook.

The data view card updates to show that the view is ready to be analyzed as it shows a new button with text **Analyze in Notebook**.

9. Choose **Analyze in Notebook** to access data in the data view in the integrated notebook environment.

> **Note**
>
> Starting up the FinSpace notebook environment for the first time may take 10-15 minutes. This is a one-time delay.

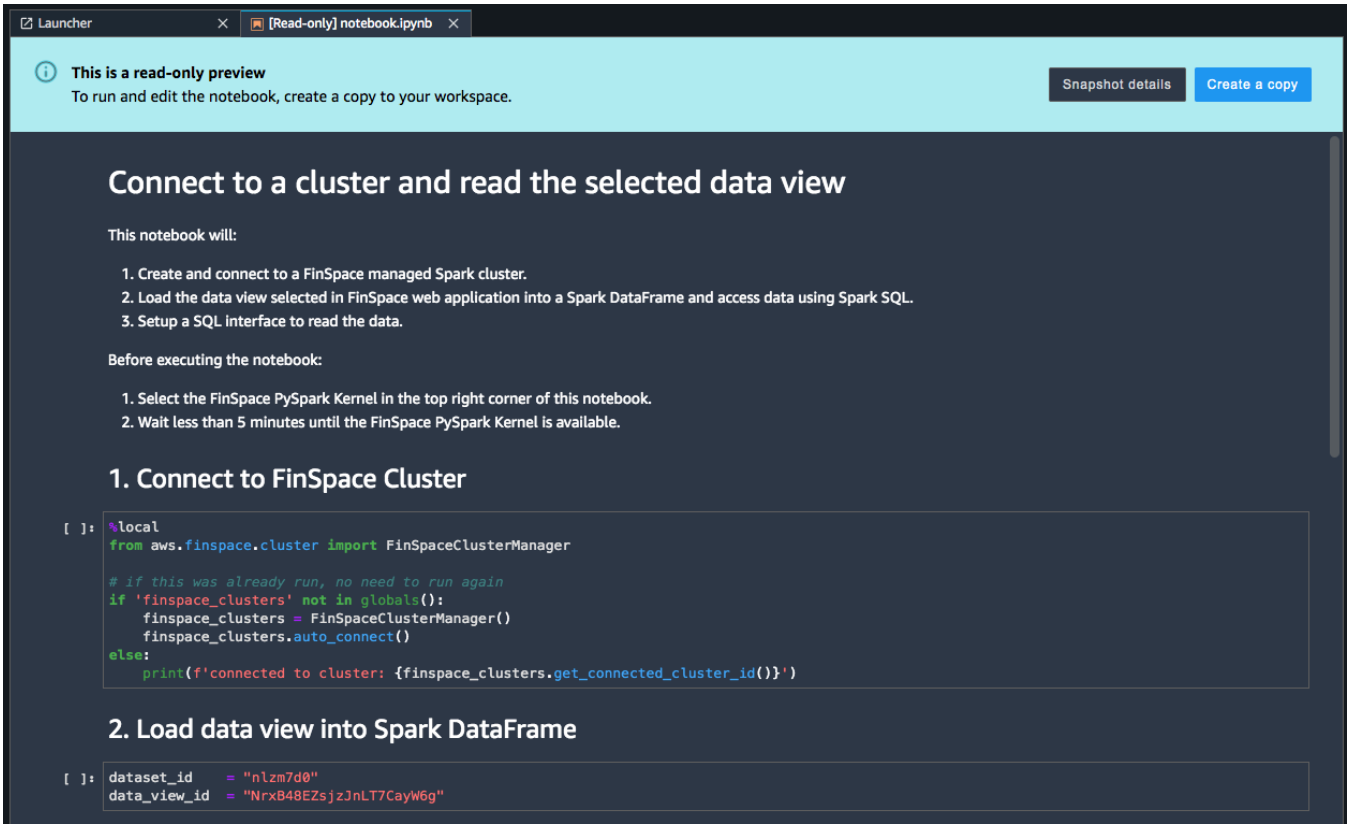
## Analyze the data view in Amazon FinSpace notebook

Before you proceed with this section, we recommend that you begin by reading Working with Amazon FinSpace notebooks.

**To analyze the data view in FinSpace notebook**

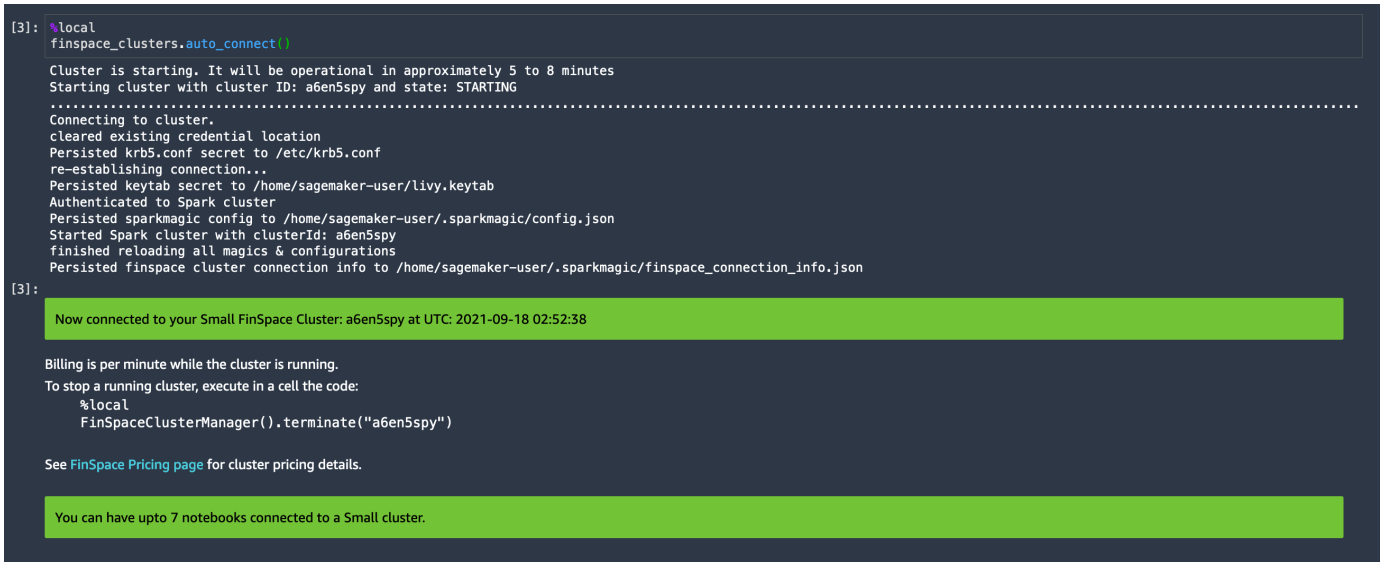
1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. Open data view in a notebook. For more information, see Opening the notebook environment.
3. A default notebook in read-only preview is populated with the details of the view. Choose **Create a copy.** The notebook is created with name **notebook.ipynb.** The notebook contains code for:

   - Starting a Spark cluster.
   - Loading the data view in a Spark DataFrame.
   - Print the schema and contents of the DataFrame.

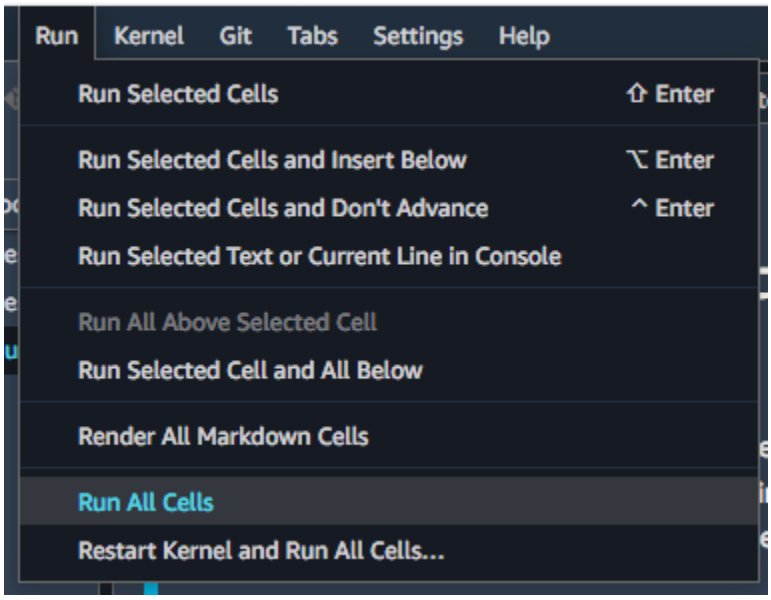



> **Note**
>
> If the kernel is starting for the first time, expect a one-time delay of approximately 5-7 minutes. The **FinSpace PySpark** kernel and a notebook instance is automatically selected.

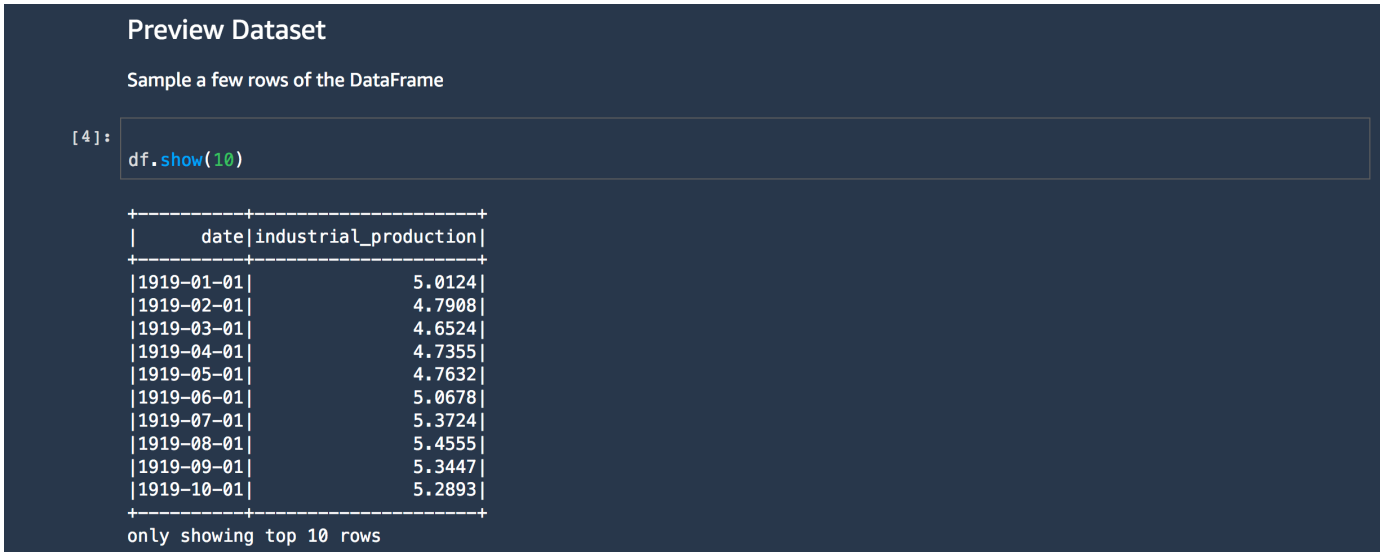
4. Start the Spark cluster by running the first cell of the notebook. Spark cluster creation takes about 5-8 minutes. If a Spark cluster is already created, then the notebook will detect the cluster and connect to it.




5. On the top menu bar, choose **Run** and then choose **Run all the cells**.




6. The executed code shows the contents of the data view.



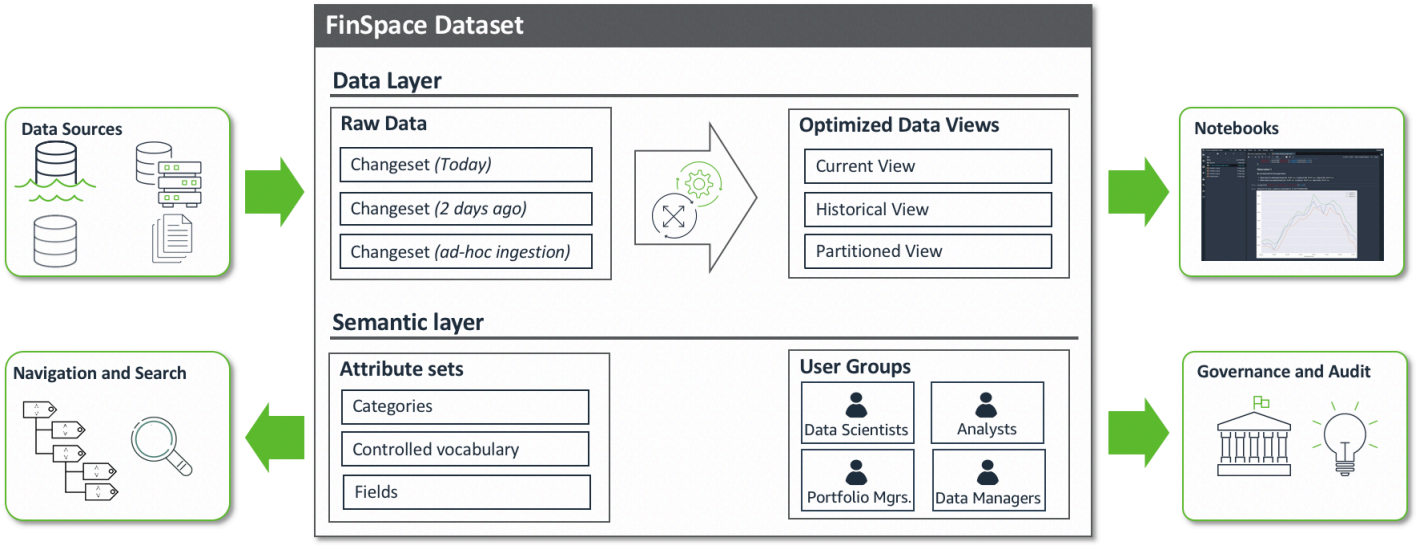

# Adding and managing data in Amazon FinSpace

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

People with different roles such as Analyst, Data Scientist, Data Engineer, Data Governor, Audit personnel use Amazon FinSpace for data organization, governance, preparation, and analysis. FinSpace supports data of any file format with additional features for structure data formats such as CSV.

FinSpace represents data in the catalog using a structure called a Dataset. Dataset is a logical container of semantically identical data and schema.




The first step is loading data into FinSpace, often referred to as ingesting data. FinSpace supports loading data in a variety of data formats and sources. You can load data by connecting in your data feeds or upload ad-hoc data through the web application.

After your data is available in FinSpace, you can do the following:

- Describe datasets to provide business context by using fields specified from Attribute Sets.
- Control who can access the data by assigning permissions to permission groups.
- Create data views that allow users to query data in FinSpace notebooks.
- Using the notebooks, create derived data by joining data and from the results of analysis of a dataset.
- Generate audit report on activity.

### Topics

- Loading data into Amazon FinSpace
- Supported data types and file formats in Amazon FinSpace
- Working with datasets in Amazon FinSpace

# Loading data into Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Data can be loaded into FinSpace from the following sources

- Amazon S3
- On-premises data stores
- Local desktop

Data can be loaded using following methods

- FinSpace web application
- SDK to connect your data feeds

# Supported data types and file formats in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace provides support for a variety of data types in structured data and file formats.

## Supported column types and values for structured data

FinSpace currently supports the following data types for the columns of structured data

- String
- Char
- Integer
- Tiny Integer
- Small Integer
- Big Integer
- Float
- Double
- Date. Supported Date format is yyyy-MM-dd. For example, 2016-12-31
- Datetime. Support Datetime format is yyyy-MM-dd HH:mm:ss. For example, 2016-12-31 15:30:00
- Boolean
- Binary

## Supported file formats

Files of any format can be ingested into FinSpace, but data view creation is only supported for the following formats:

- CSV – Only UTF-8 encoding is supported
- JSON
- Parquet
- XML

## Format options for loading data

FinSpace supports following formatting options when loading data in supported formats types. Currently, the only formats that FinSpace supports are CSV, JSON, Parquet, and XML.

> **Note**
>
> The FinSpace web application only supports ingestion for CSV format for creation of data views and comma delimited and `withHeader` option. Other formats are supported with SDK.

## CSV

This value designates comma-separated-values as the data format (for example, see RFC 4180 and RFC 7111).

You can use the following `formatParams` values with `FormatType="csv"`:

1. `separator` – Specifies the delimiter character. The default is a comma "," but any other character can be specified.
2. `escaper` – Specifies a character to use for escaping. This option is used only when reading CSV files. The default value is none. If enabled, the character that immediately follows is used as-is, except for a small set of well-known escapes (\n, \r, \t, and \0).
3. `quoteChar` – Specifies the character to use for quoting. The default is a double quote ("). Set this to -1 to disable quoting entirely.
4. `multiLine` – A Boolean value that specifies whether a single record can span multiple lines. This can occur when a field contains a quoted new-line character. You must set this option to "True" if any record spans multiple lines. The default value is "False", which allows for more aggressive file-splitting during parsing.
5. `withHeader` – A Boolean value that specifies whether to treat the first line as a header. The default value is "True".
6. `skipFirst` – A Boolean value that specifies whether to skip the first data line. The default value is "False".

> **Note**
>
> If any of the default values are changed, all format values must be supplied.

### JSON

This value designates a JavaScript Object Notation data format.

You can use the following `formatParams` values with `FormatType="json"`:

1. `jsonPath` – A JsonPath expression that identifies an object to be read into records. This is particularly useful when a file contains records nested inside an outer array. For example, the following JsonPath expression targets the id field of a JSON object.

```
format="json", format_options={"jsonPath": "$.id"}
```

### Parquet

This value designates Apache Parquet as the data format.

There are no `formatParams` values for `FormatType="parquet"`.

### XML

This value designates XML as the data format, parsed through a fork of the XML data source for Apache spark parser.

You can use the following `formatParams` values with `FormatType="xml"`:

1. `rowTag` – Specifies the XML tag in the file to treat as a row. Row tags cannot be self-closing.
2. `encoding` – Specifies the character encoding. The default value is "UTF-8".
3. `excludeAttribute` – A Boolean value that specifies whether you want to exclude attributes in elements or not. The default value is "false".
4. `treatEmptyValuesAsNulls` – A Boolean value that specifies whether to treat white space as a null value. The default value is "false".
5. `attributePrefix` – A prefix for attributes to differentiate them from elements. This prefix is used for field names. The default value is "_".
6. `valueTag` – The tag used for a value when there are attributes in the element that have no child. The default is "_VALUE".
7. `ignoreSurroundingSpaces` – A Boolean value that specifies whether the white space that surrounds values should be ignored. The default value is "false".

# Working with datasets in Amazon FinSpace

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The Amazon FinSpace dataset is a logical container of semantically identical data and schema. Dataset keeps track of all the data that is ingested, and also tracks data views that get generated on the ingested data. The data views are used to access the data for analysis within notebooks. Dataset can contain structured data with a schema or unstructured data like PDF files or blobs.

**Topics**



## Dataset details page

**Important**

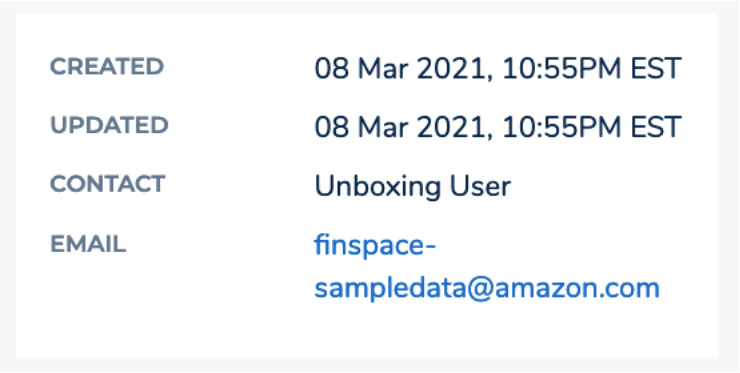
Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The dataset details page contains detailed information about the dataset. This page contains overview of a dataset, all the data views created for the dataset, the schema and permissions related to a dataset under the following tabs.

- Data Overview
- All Data Views
- Schema
- Permissions

From the right side of the page, you can edit the dataset description or remove the dataset by choosing the **More** menu.

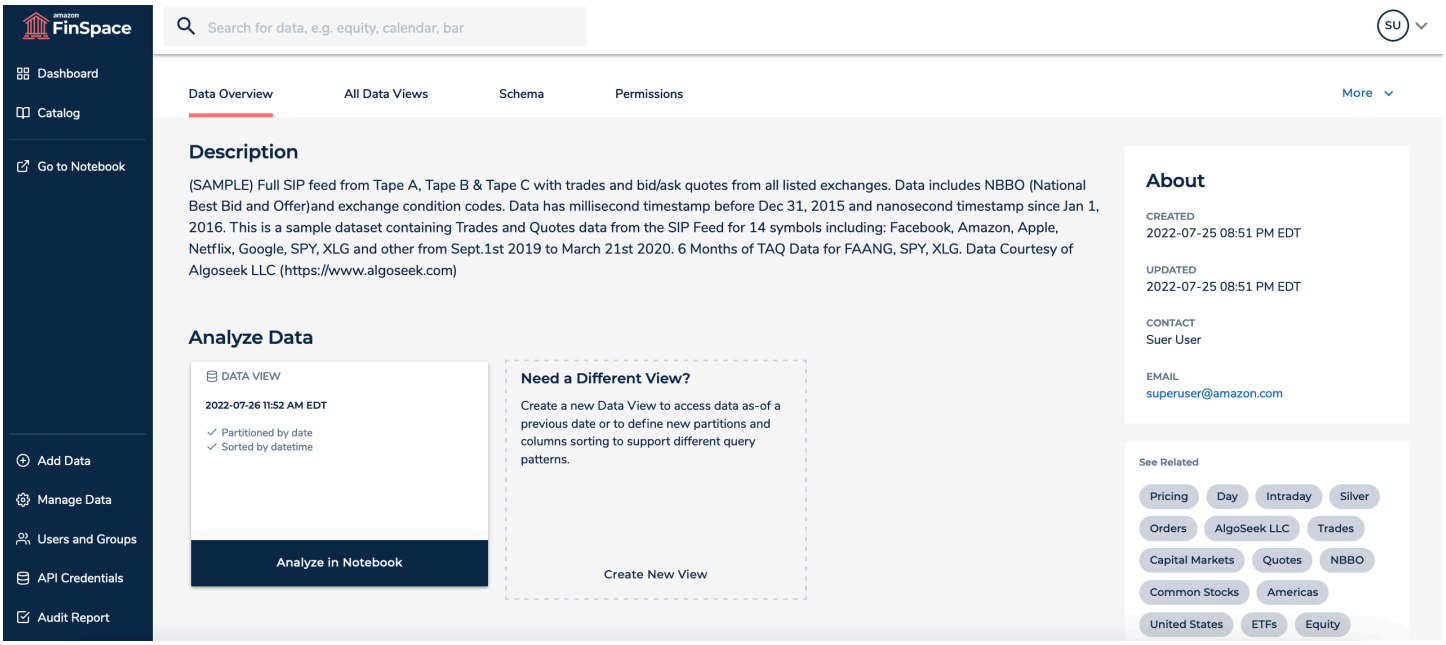
You can also view the information related to when the dataset was created and the user who created this dataset.




From the **See Related** section, you can easily navigate to related datasets in the application. Each label in this section corresponds to attribute values and category values associated to a dataset. The labels listed in this section match the values of the attributes that you select at the bottom of the **Data Overview** tab. Selecting any labels will take you to the data browser where other datasets with the same label will be shown in the results.

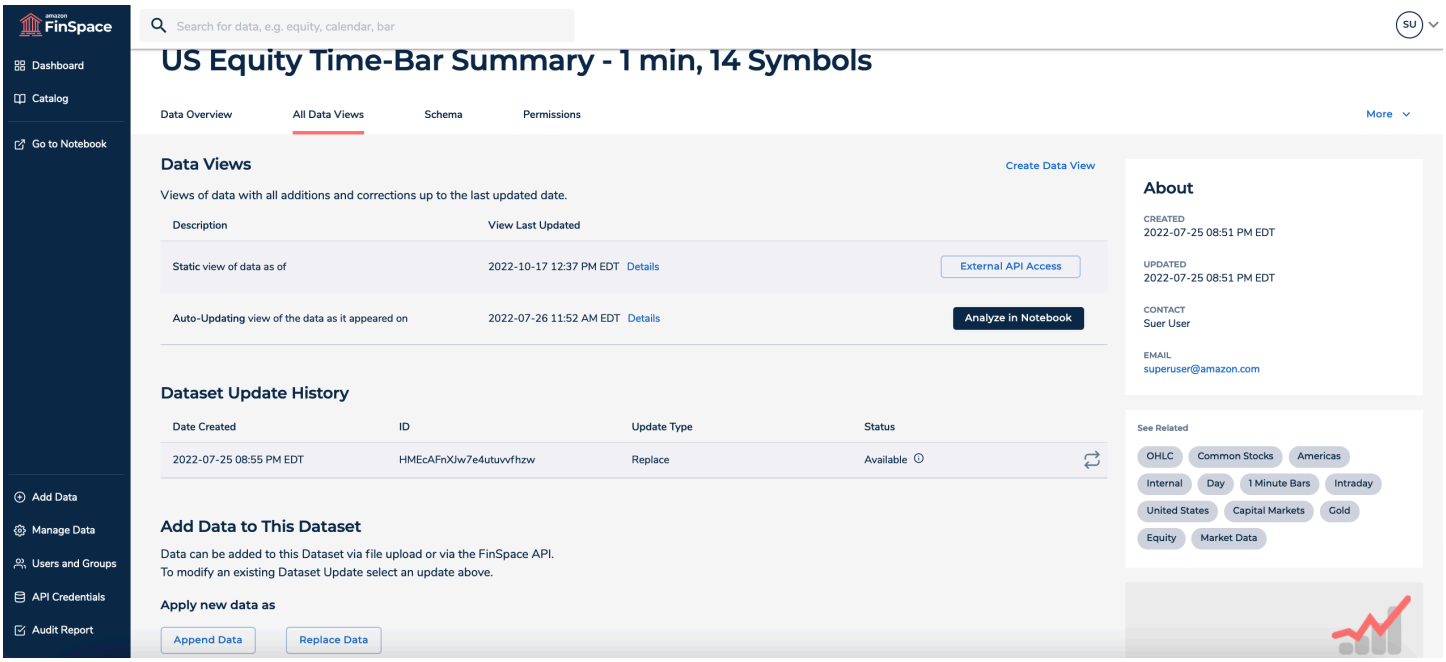
## Data Overview

This tab shows the description of the dataset, latest data views, and associated attribute sets that describe the dataset.




## All Data Views

This tab shows the details of all the data that is ingested into the dataset as changesets, and all the data views that have been created.

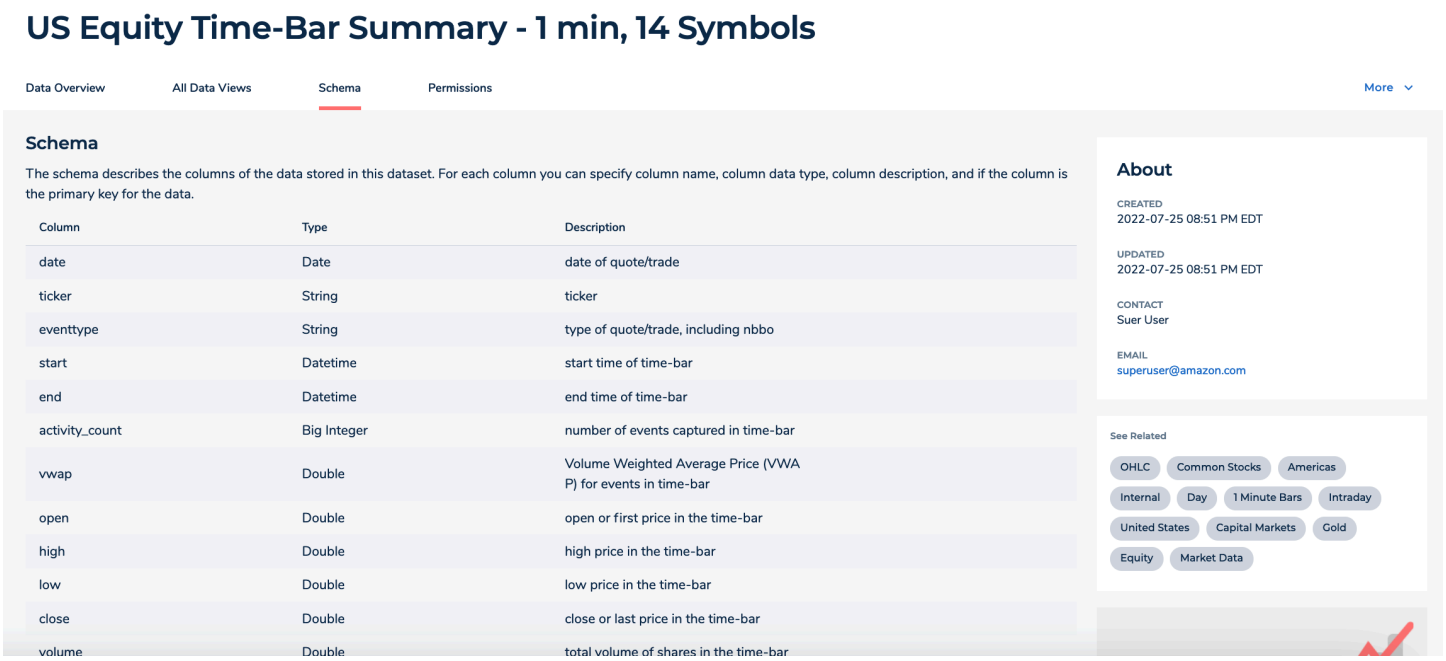


In this tab, you can do the following:

- View list of data views under the **Data Views** section. Choose **Details** to view detailed information about a specific data view.
- Load a data view for analysis. Choose the **Analyze in Notebook** button to open the data view in FinSpace notebook. Choose the **External API Access** button to access the data view externally using the FinSpace API.
- Create new data views by choosing the **Create Data View** button. For more information, see Create data view.
- View dataset update history and make corrections to datasets.
- Load data to the dataset by uploading a file or through FinSpace API.
- Create changeset with `Append` and `Replace` type. For more information, see Creating changesets in a dataset.

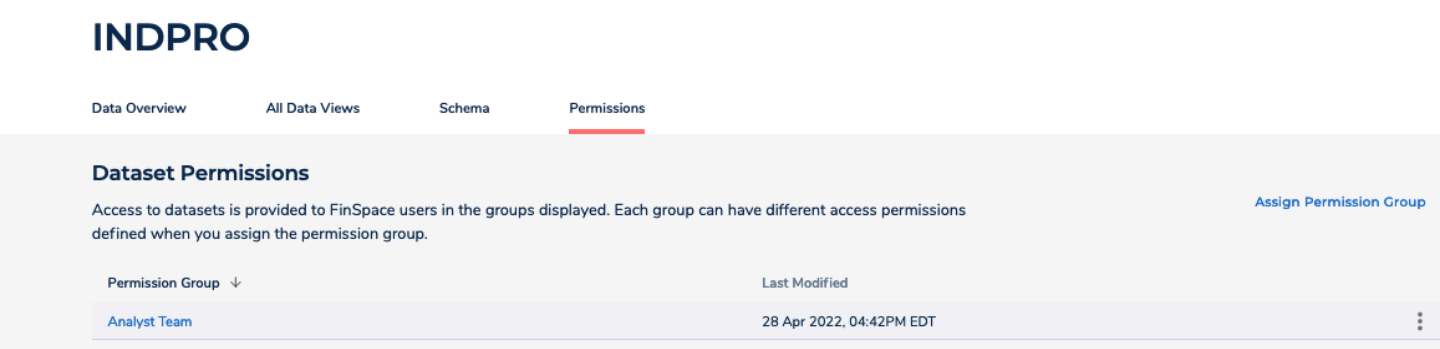
### Schema

This tab shows the schema of the dataset. The existing schema can only be edited if no data views have been created.




### Permissions

This tab shows the list of permission groups that are entitled to use the dataset. From this section, you can assign new permission groups to the dataset by choosing **Assign Permission Group**.




## Creating a dataset

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

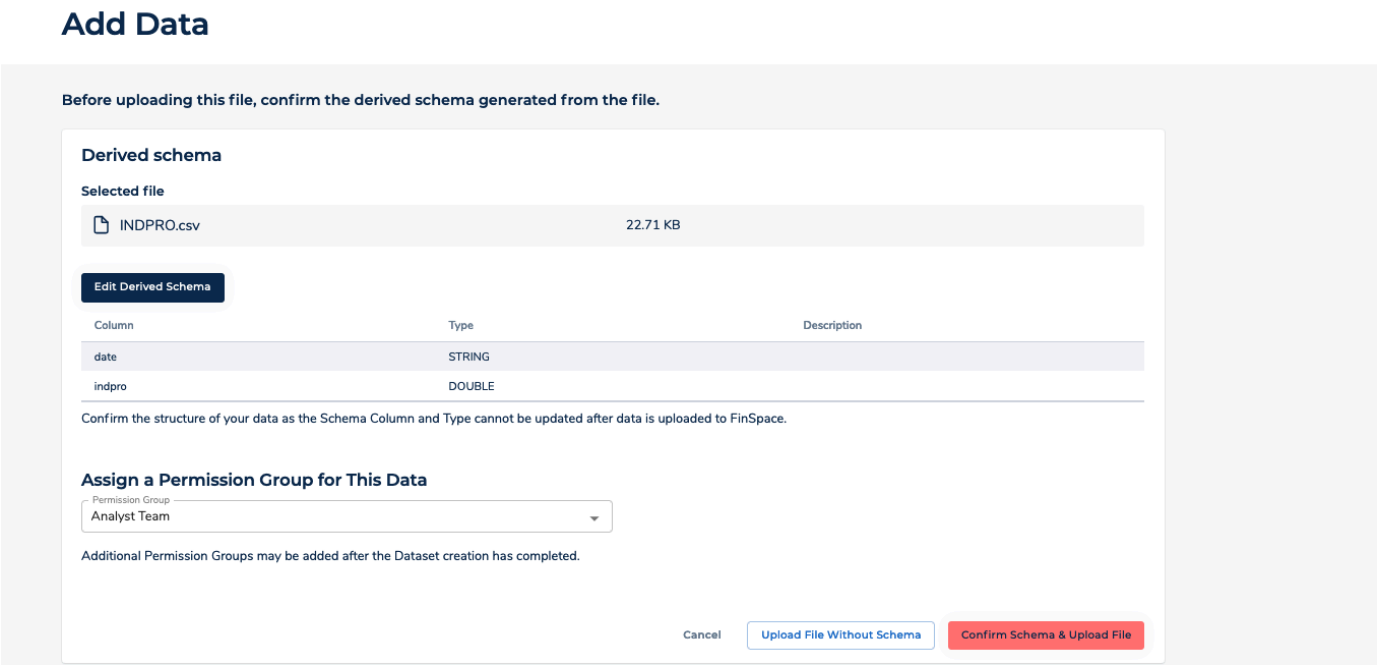
**Note**

In order to create and manage datasets, you must be a superuser or a member of a group with necessary permissions – **Create Datasets**.

A dataset can be created by loading a file using the Amazon FinSpace web application.

### To create a dataset

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Add Data**.
3. Drag and drop a .csv file or choose **Browse Files** to select a file. Once the file is detected by the web application, schema of the file will be displayed. The column names are read from the file and data types are inferred.




4. Change the data types as required by choosing **Edit Derived Schema**. Take note of the data types and formats that are supported.
5. Choose **Save Schema**.
6. Choose **Confirm Schema & Upload File**. This action starts the following process:

    1. Create a dataset with name of the .csv file that was loaded and takes you to the dataset details page.
    2. Once the upload of the sample data file is complete, a changeset is created with the content of the data file. Verify by checking the **Dataset Update History** table under **All Data Views** tab.
    3. Data view creation process is started. Once the upload of the sample data file is complete, a process is kicked off to create a data view that can be analyzed in a notebook.

        For small files of up to 100 megabytes, data view creation takes approximately 2 minutes. For larger files of around 1 gigabyte, expect data view creation to take approximately 3-4 minutes. Views with partitioning and sorting schemes may take longer.

        Once a dataset is created, you can start adding data to it. A new set of data added to a dataset creates a corresponding changeset.

# Creating changesets in a dataset

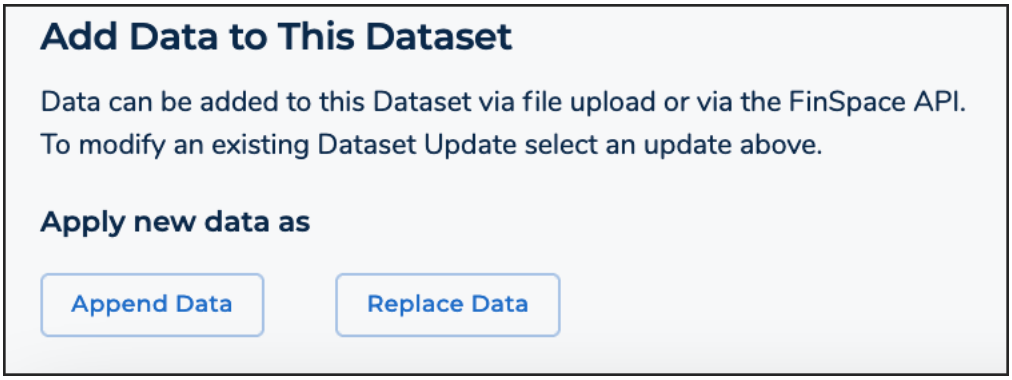
**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Data files are added to datasets and tracked as a changeset. A changeset is created in a dataset when one or more data files are ingested in a single operation. All changesets in a dataset are preserved unless a dataset itself is deleted. A changeset is created with a unique identifier and a system timestamp is assigned to it at the time of creation.

**A changeset is created as one of the following types**

- **Append** – New changeset is considered an addition to the end of the prior ingested changesets. For example, addition of a new daily file.
- **Replace** – New changeset is considered a replacement to all prior ingested changesets in a dataset. This does not mean that the prior ingested changesets are deleted but they will not be considered for the view creation.




**Replace data**

**To create a changeset with type as Replace**

1. From the homepage search for a dataset where you want to replace data.

2. Choose the dataset name to view the dataset details page.
3. Choose the **All Data Views** tab.
4. Scroll down and choose **Replace Data**.
5. Choose **Select CSV File** to select and upload a file from your desktop.
6. Once the file is uploaded, choose the input format for the ingested data from the following options:
   - **Delimiter** – Specifies the delimiter character. The default value is *Comma*.
   - **Escape Character** – Specifies a character to use for escaping. The default value is *None*.
   - **Quotes** – Specifies the character to use for quoting. The default value is *Double Quotes* (").
   - **Multiline Records** – Specifies whether a single record can span multiple lines. By default this option is disabled. Enable this option if you want any record to span multiple lines.
   - **Treat First Line As Header** – Specifies whether to treat the first line as a header. By default this option is disabled.
   - **Skip First Data Line** – Specifies whether to skip the first data line. By default this option is disabled.
7. Choose **Replace Data**.
8. Once the file upload is complete, you should see a new entry for a changeset of type *Replace* under the **Dataset Update History** table with a **Pending** status. Once the status is set to **Available**, a data view that includes the new changeset can be created.

### Append data

#### To create a changeset with type as Append

1. From the homepage, search for the dataset to which you want to append data.
2. Choose the dataset name to view the dataset details page.
3. Choose the **All Data Views** tab.
4. Scrolls down and choose **Append Data**.
5. Choose **Select CSV File** to select and upload a file from your desktop.
6. Once the file is uploaded, choose the input format for the ingested data from the following options:
   - **Delimiter** – Specifies the delimiter character. The default value is *Comma*.
   - **Escape Character** – Specifies a character to use for escaping. The default value is *None*.

- **Quotes** – Specifies the character to use for quoting. The default value is *Double Quotes* (").
- **Multiline Records** – Specifies whether a single record can span multiple lines. By default this option is disabled. Enable this option if you want any record to span multiple lines.
- **Treat First Line As Header** – Specifies whether to treat the first line as a header. By default this option is disabled.
- **Skip First Data Line** – Specifies whether to skip the first data line. By default this option is disabled.

7. Choose **Append Data**.
8. Once the file upload is complete, you should see a new entry for a changeset of type *Append* under the **Dataset Update History** table with a **Pending** status. Once the status is set to **Available**, a data view that includes the new changeset can be created.

## Corrections to a dataset

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

A changeset can be ingested as a correction to an already created changeset. This action does not delete the prior ingested set but signifies that the replaced changeset will be used when a view is created if both changesets fall under the specified date and time of the view.

**To create a changeset that is a replacement to an existing changeset**

1. From the homepage, search for the dataset that you want to make corrections to.
2. Choose the dataset name to view the dataset details page.
3. Choose the **All Data Views** tab.
4. Under the **Dataset Update History** table, from the list of changesets identify the changeset to be replaced and then choose the corrections icon (  ).

5. Choose **Choose CSV File** to select and upload a file from your desktop.
6. Once the file is uploaded, choose the input format for the ingested data from the following options:
   - **Delimiter** – Specifies the delimiter character. The default value is *Comma*.
   - **Escape Character** – Specifies a character to use for escaping. The default value is *None*.
   - **Quotes** – Specifies the character to use for quoting. The default value is *Double Quotes* (").
   - **Multiline Records** – Specifies whether a single record can span multiple lines. By default this option is disabled. Enable this option if you want any record to span multiple lines.
   - **Treat First Line As Header** – Specifies whether to treat the first line as a header. By default this option is disabled.
   - **Skip First Data Line** – Specifies whether to skip the first data line. By default this option is disabled.
7. Choose **Save**. The changeset is added to the **Dataset Update History** table with a **Pending** or **Running** status that changes to **Available** once the update is successful.

## Removing a dataset

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

A dataset can be permanently removed from your Amazon FinSpace environment.

**To remove a dataset**

1. From the homepage, search for the dataset that you want to remove.
2. Choose the dataset name to view the dataset details page.
3. On the top-right corner, choose the **More** menu and then choose **Remove Dataset**.
4. In the confirmation dialog box, choose **Remove Dataset**.

> **Note**
>
> This action is irreversible. Once removed, a dataset cannot be recovered.

# Data connectors in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

With data connectors, you can access external data sources from within your Amazon FinSpace environment. Each data connector is associated with a *data provider*, which defines the type of external data source.

FinSpace currently supports the following provider:

- **Goldman Sachs Financial Cloud for Data provider** – Enables access to data from the Goldman Sachs Financial Cloud for Data, a cloud-native modular collection of services. It delivers enhanced financial market analytics by providing access to market data, reference data, research, and a variety of data products curated by Goldman Sachs and mapped to a unified data model.

**Topics**

# Tutorial: Creating a connector for Goldman Sachs Financial Cloud for Data

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

This tutorial guides you through the steps to create a data connector for the Goldman Sachs Financial Cloud for Data (GSFCD) provider.

## Prerequisites

Before you proceed, make sure that you have the following available:

- **Goldman Sachs Financial Cloud for Data API credentials** – These credentials will be used to connect to the GSFCD. The credentials will be stored in AWS Secrets Manager so that the data connector can use them securely.
  - Registered users for Goldman Sachs Financial Cloud for Data can obtain new API credentials from Goldman Sachs Developer website.
  - New users can submit a request to obtain API credentials at Goldman Sachs Financial Cloud for Data.
- **A FinSpace environment** – You can only use a data connector in the FinSpace environment where it was created. For more information, see Create an Amazon FinSpace environment.

## Step 1: Add connector details

### To add connector details

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, choose **Data Providers**.

> **Tip**
>
> Alternatively, you can also perform the following steps:
>
> 1. In the left pane, choose **Environments**.
> 2. From the list of environments, choose the name of the environment where you want to create a data connector.
> 3. On the environment details page, scroll down to **Data Connectors** and choose **Create connector**. The **Data Providers** page opens.

3. On the **Data Providers** page, for the **Goldman Sachs Financial Cloud for Data** provider, choose **Add connector**.
4. On the **Connector details** page, provide a unique **Connector name**, and choose an account with superuser to run the connector.
5. For **Scheduled runs**, select this option if you want to schedule automatic connector runs. The data connector will run daily at 00:00 UTC.

   Clear this option if you don't want to schedule automatic runs. You will need to manually start the data connector run from the console. For more information, see Running a data connector.
6. Choose **Next** and proceed to Step 2: Add a secret name.

## Step 2: Add a secret name

FinSpace uses AWS Secrets Manager to store the API credentials that your FinSpace environment will use to connect to the Goldman Sachs Financial Cloud for Data API. For more information, see Secrets Manager concepts in the *AWS Secrets Manager User Guide*.

When you choose **Next** on the **Connector details** page in the previous step, the **Secret name** page opens. You can choose an existing secret name or create a new one.

**To add a secret name**

1. On the **Secret name** page, choose an existing secret name from the dropdown list.
2. You can also create a new secret name on this page by choosing the **Create new secret** option from the list.

   a. Under the **Create new secret** section, for **Secret name**, enter a unique name for the secret.

b. Enter the key-value pair for your secret in **Client ID** and **Client secret**, respectively.

c. Choose an encryption AWS KMS key. This key will be used by AWS Secrets Manager to encrypt your secret. You can select an existing KMS key from the dropdown or create a new one by using the AWS Key Management Service. For more information, see the AWS Key Management Service Developer Guide.

> **Note**
>
> By default, this field displays the KMS key that you used to create the environment where you're creating this data connector.

3. Choose **Next** and proceed to Step 3: Add customer IAM role.

> **Note**
>
> You can also create a secret directly from the AWS Secrets Manager console. For more information, see Create a secret in the *AWS Secrets Manager User Guide*.

## Step 3: Add customer IAM role

In FinSpace, you can securely control access to data connectors by creating IAM policies and attaching them to roles. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal uses an IAM entity (user or role) to make a request. For more information, see Roles terms and concepts in the *IAM User Guide*.

When you choose **Next** on the **Secret name** page in the previous step, the **Customer IAM role** page opens. You can select an existing role or create a new one.

**To add a customer IAM role**

1. On the **Customer IAM role** page, choose an existing role ARN from the dropdown list.
2. You can also create a new role on this page by choosing the **Create new customer IAM role** option from list.

   First create a permissive IAM policy and then create an IAM role. Then attach the new policy to it.

**To create an IAM policy**

a. Under the **Create a policy** section, choose **Copy code** to copy the policy code. You will use this code to create an IAM permissions policy.

b. Choose **Go to policy creation form**. This button opens the **Create policy** page in a new tab.

> **Note**
>
> Do not close the **Customer IAM role** tab.

c. On the **Create policy** page, choose the **JSON** tab. Delete any prepopulated JSON code, and then paste the policy code that you copied in previous step.

d. Choose **Next: Tags**. (Optional) Add metadata to the policy by attaching tags as key-value pairs.

e. Choose **Next: Review**.

f. On the **Review policy** page, enter a **Name** and a **Description** (optional) for the policy that you're creating. Review the policy **Summary** to see the permissions that are granted by your policy. Then choose **Create policy** to save your work.

> **Note**
>
> Remember this policy name because you will need it while creating a role.

**To create an IAM role**

1. Return to the **Select customer IAM role** tab. Under the **Create a customer IAM role** section, choose **Copy code** to copy the trust relationship code.
2. Choose **Go to customer IAM role form**. This button opens the **Create role** setup in a new tab.

> **Note**
>
> Do not close the **Customer IAM role** tab.

3. On the **Select trusted entity** page, for **Trusted entity type**, choose **Custom trust policy**.

4. Under the **Custom trust policy** section, delete any prepopulated code, and then paste the trust relationship code that you copied in the previous step.
5. Choose **Next**.
6. On the **Add permissions** page, for **Permissions policy**, search for the policy name that you created in step f in "To add a customer IAM role". Select the policy check box and choose **Next**.
7. On the **Name, review, and create** page, add a role name. Review the policy and permission details and choose **Create role**.

   > **Note**
   >
   > Remember this role name because you will need it in the next step.

3. Return to the **Select customer IAM role** tab. For **Customer IAM role**, enter the name of the role you created in the previous step.
4. Choose **Next** and proceed to Step 4: Review and create.

> **Note**
>
> You can also create the IAM role and policy directly from the AWS Identity and Access Management console. For more information, see Creating an IAM role (console) in the *IAM User Guide*.

## Step 4: Review and create

Review the connector details, secret name, and customer IAM role, and then choose **Create connector**.

After the new data connector is created, the connector details page opens where you can perform other operations using a data connector. To verify that the new connector setup is complete, see the **Connector summary** section and ensure that the **Status** is **Active**. The connector will start syncing automatically when it's connected. For more information, see Connector details.

**Note**

- If you create multiple GSFCD data connectors for a single Amazon FinSpace environment, duplicate datasets are created in FinSpace if the GSFCD client access credentials that you use have an overlap in the datasets they have access to. To avoid this, only create multiple connectors with credentials that don't have overlapping access to datasets.
- Datasets that are created when a GSFCD connector runs are placed in a system-generated permission group. You can't add them to other permission groups.

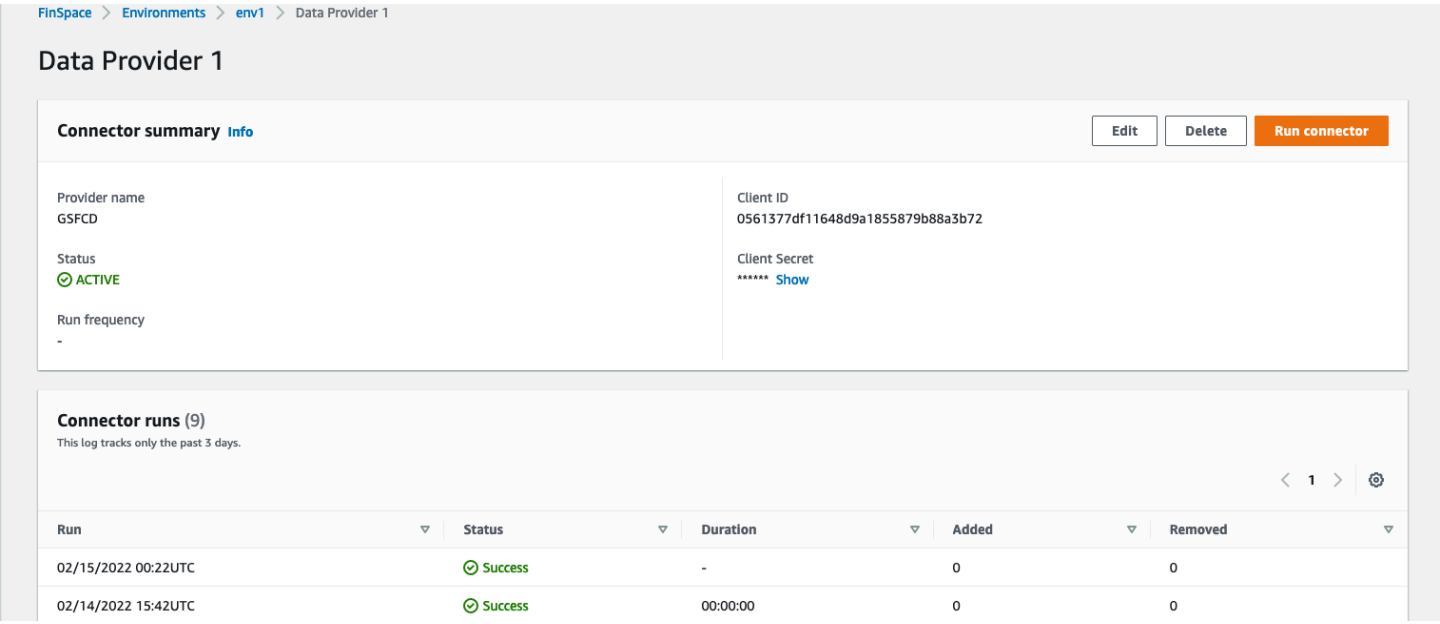
## Connector details

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The connector details page displays a summary of details for each data connector. It consists of two sections:

- **Connector summary** – This section displays details of the connector that you created, such as the provider name, status of the connector, and run frequency. In this section, you can also edit, delete, or run connectors.
- **Connector runs** – This section displays the date, status, and duration of each data connector run in a table. The table shows logs for only the past three days.




**Note**

- The connector summary displayed on this page might differ for each data provider.
- Superusers automatically have access to all datasets that a connector creates.

## Running a data connector

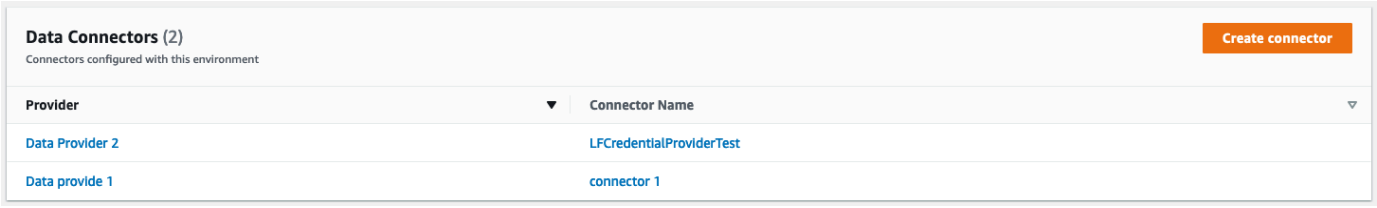
After you've created a data connector, you can run it from the connector details page. When a data connector runs, it retrieves all the datasets from the provider and populates them as datasets into the FinSpace web application, which can be accessed with the provided credentials. All datasets created by running a connector are placed in a FinSpace permission group with naming convention as `<Connector Name> Group (System Created)`. You can assign users to this permission group to grant them access.

**Note**

You can only use a data connector in the environment where you create it.

### To run a data connector

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, choose **Environments**.
3. From the list of environments, choose the name of the environment where you created the data connector.
4. On the environment details page, scroll down to **Data Connectors** and choose the name of the data connector that you added.




5. On the **Connector summary** page, choose **Run connector**. The status is updated under the **Connector runs** section.

> **Note**
>
> - The run operation could take about three to five minutes to complete.
> - When a data connector run is still in progress, the **Edit**, **Delete**, and **Run connector** buttons are disabled.

After you get a confirmation message, the data connector connects to the data provider and loads the available datasets into the FinSpace web application. For more information about using datasets in the FinSpace web application, see Using external datasets in Amazon FinSpace.

## Editing a data connector

### To edit a data connector

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.

2. In the left pane, choose **Environments**.
3. From the list of environments, choose the name of the environment where you created the data connector.
4. On the environment details page, scroll down to **Data Connectors** and choose the name of the data connector that you want to edit.
5. On the **Connector summary** page, choose **Edit**. The **Edit connector** page opens, and you can edit the details as required.

> **Note**
>
> - You can't edit the following fields:
>   - **Environment**
>   - **Data provider**
>   - **Connector name**
> - For Goldman Sachs Financial Cloud for Data connectors, if you change the **secret name**, you must modify the IAM role.

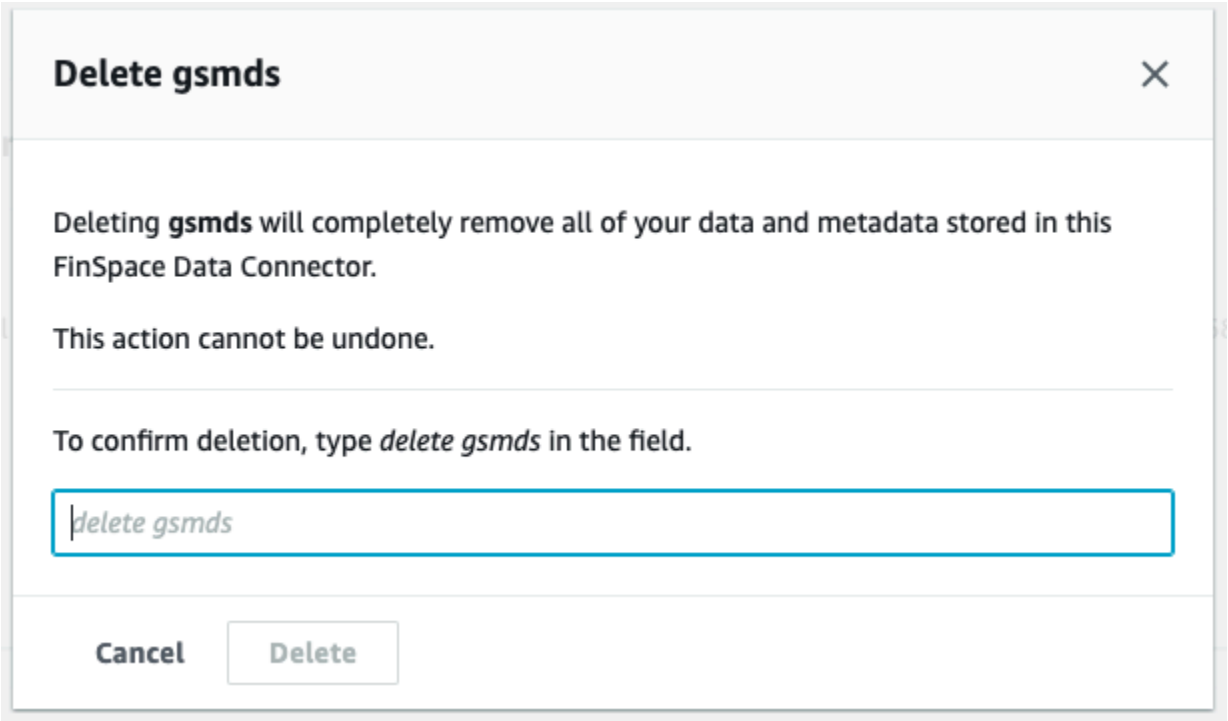
## Deleting a data connector

> **Note**
>
> This action is irreversible. Deleting will completely remove all of your datasets and associated metadata that the data connector creates in the FinSpace environment.

### To delete a data connector

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. In the left pane, choose **Environments**.
3. From the list of environments, choose the name of the environment where you created the data connector.
4. On the environment details page, scroll down to **Data Connectors** and choose the name of the data connector that you want to delete.

5. On the **Connector summary** page, choose **Delete**.
6. On the confirmation dialog box, enter the name of the connector to delete it.




**Note**

The following entities that are automatically created by a data connector remain in your FinSpace environment, even after you delete the data connector. You can later remove these entities manually if you choose to.

- Permission groups.
- Categories – After deleting a data connector, the categories are still available under the **External Data** categories in the data browser and the **Categories** page.
- Attribute sets – After deleting a data connector, these attributes are still available under the **External Data Attribute Set** section in the **Attribute Sets** page.

# Using external datasets in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

External datasets are the datasets that a data connector creates. They refer to the data that resides outside of FinSpace. You can use external datasets to discover and use data assets that reside in an external data repository from within FinSpace—without having to copy the data into a FinSpace environment.

## External datasets in the FinSpace catalog

You can access the external datasets by using the FinSpace web application. Currently, FinSpace only supports the Goldman Sachs Financial Cloud for Data as an external datasets source. In the catalog, these datasets have the *Attribute-set* label of **Goldman Sachs Financial Cloud for Data** applied to them.

The external datasets behave like regular datasets, but are different in the following ways:

- You can't delete external datasets. FinSpace can only remove them as a result of a data connector run.
- You can't add data to an external dataset from FinSpace using changesets. For this reason, the **Add Data** and **Replace Data** buttons aren't visible when you view an external dataset in the FinSpace catalog.
- Each dataset contains a system-generated data view called an *external data view*. You can't generate any other additional data views.
- You can't add or remove external datasets from the system-generated permission groups where they were added by the data connector. To grant users access to external datasets, add them to the system-generated permission groups.
- You can't add external datasets to permission groups other than the system-generated one that is created by running the data connector.

- You can't remove the system-generated attribute set that is applied to an external dataset.

## Browse external datasets

### To browse external datasets using the FinSpace catalog

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Catalog**.
3. From the data browser, choose the **External Data** category.
4. Choose the **Goldman Sachs Financial Cloud for Data** category, and then select the required dataset. The dataset details page opens, and you can view details about the selected dataset.

## Access external datasets from FinSpace notebook using Spark

The process of accessing external data using a FinSpace notebook is same as accessing any other datasets. For more information, see Access datasets from a notebook.

### Specifying additional parameters

With external datasets, you can also specify additional parameters to pre-filter the data that returns to a Spark DataFrame. To do this, you use the *partition_filter* parameter. The parameters that you specify depend on the particular data provider that you use. For information on the specific parameters for the Goldman Sachs Financial Cloud for Data, refer to the Marquee documentation.

The following is an example of specifying additional parameters.

```
        df = analytics.read_data_view( dataset_id="<dataset_id>",
 data_view_id="<data_view_id>",
partition_filter={
"exchange": ["NASDAQ", "NYSE"],
"symbol": ["AMZN", "GOOG"],
}
)
```

In the preceding example, the <dataset_id> is a FinSpace dataset ID such as *rgg1nj1*, and <data_view_id> is a FinSpace data view ID such as *VrvKEKnA1El2nr821BaLTQ*.

# Data views for querying data

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Data views provide you access to the data stored in Amazon FinSpace to perform queries. Each data view represents a picture of the content of a dataset at a given point in time. A data view can be historically created from a specified data, or can be auto-updated as new data is ingested for the dataset via changeset. Multiple views can be created from a dataset with different dates or with different partitions and column sorting.

**Topics**



## Data view concepts

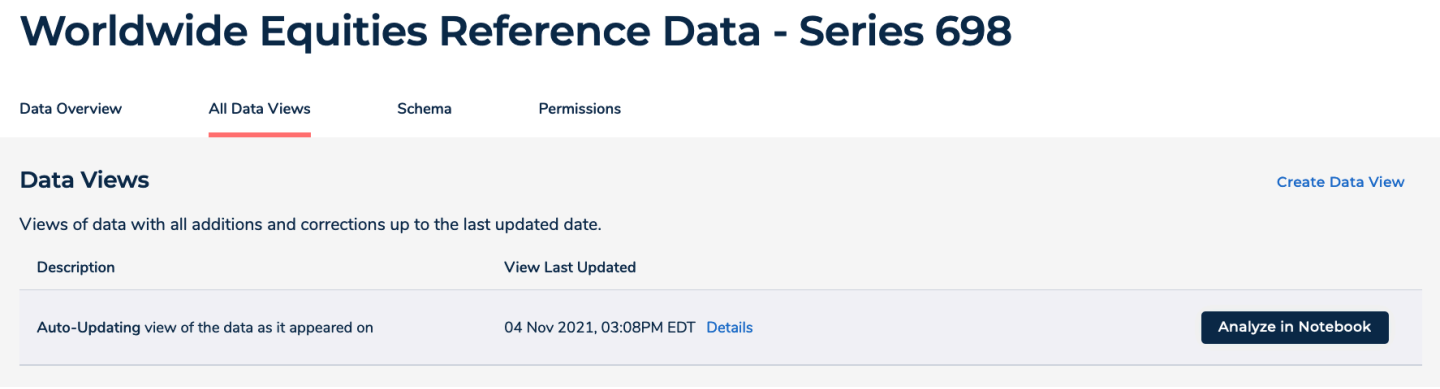
**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

### View Types

Two types of views can be setup for a dataset:

- **Auto-Update view** – A data view with all additions (Append) and corrections (Replace, Replace Changesets) for a dataset. Future additions and corrections to this dataset are automatically applied to this view.
- **Static view** – A data view with all additions (Append) and corrections (Replace, Modify) up to a specified date and time for creation of view i.e. the view will be constructed from only those changesets that were created before the specified time and date. No future additions or corrections will be applied to this view.




A data view is constructed from changesets. Two factors are taken into account for the changesets to be considered in a view:

1. Specified date and time to create the view – All the changesets created prior to the specified date and time are considered for the view. In case of an auto-update view, the specified date and time is current day and timestamp.
2. The changeset types are interpreted for a creating a data view in the following ways:
   - **Changeset with Append type** – Changeset is interpreted as an addition to the end of all the prior created changesets. The changeset will be considered for view creation.
   - **Changeset with Replace type** – Changeset is interpreted as a replacement to all prior created changesets. No changesets created before a changeset of this type are considered for the view creation.
   - **Changeset created as a correction** – Changeset is interpreted as a replacement to a specific prior created changeset. The prior created changeset will be not considered for the view creation.

**View Last Updated**

The timestamp represents the point in date and time for which the view is created. For static view, it will be the timestamp that you specified at the creation of the view. For auto-update view, it will be the last time it was updated when a new changeset was added.

**Data View ID**

The unique identifier for a data view.

**Dataset ID**

The unique identifier for a dataset.

**Data Access**

A view can be prepared to be accessed and used in:

- FinSpace notebook using integrated Spark clusters.
- Externally via the FinSpace API. The format of this view can be customized by specifying file format, delimiter, compression type.

**Partitioning**

Partitioning can be configured to optimize queries.

**Sorting**

The data in the data view can be sorted by one or more columns. Sorting data helps with query performance.

## Create data view

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

> **Note**
>
> In order to create and manage data views, you must be a superuser or a member of a group with necessary permissions – **Create Datasets**.

**To create a data view**

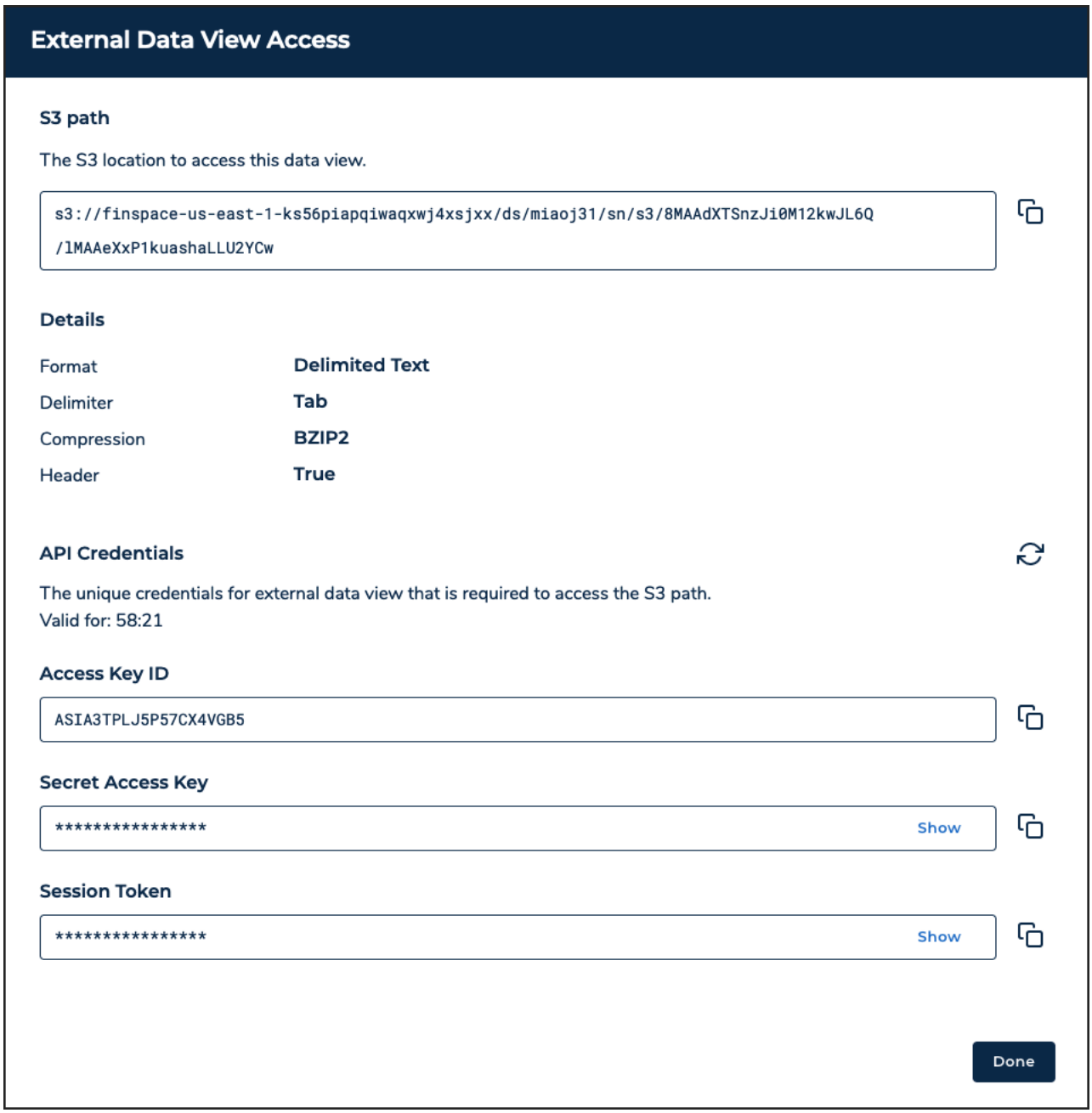
1. From the homepage, search the dataset for which you want to create a view.
2. Choose the dataset name to view the dataset details page.
3. Choose the **All Data Views** tab.
4. Choose **Create Data View**.
5. On the **Create a New View** page, choose a **View Type**. To know more about the view types, see Data view concepts.
6. For **Access Through**, choose one of the following options:
   - **FinSpace notebook using Spark** – For accessing the data view using the notebook.
   - **Externally using FinSpace API** – For accessing the data view using API. When you choose this, other fields to specify the input format are displayed.
7. Select partitioning for the data view under **Partition Data**.
   - If partitioning is not required, choose **No**.
   - If partitioning is required, choose **Yes**.
     - Select the columns to partition the view.
     - After you have selected the columns, **Order Partitions** list is populated on the right side of the menu with the selected fields. Drag the selected columns to control the partition order.
8. Select sorting for the view.
   - If sorting is not required, choose **No**.
   - If sorting is required, choose **Yes**.
     - Select the columns to sort the view.
     - After you have selected the columns, **Column Sort-By Order** list is populated on the right side of the menu with the selected fields. Drag the selected columns to control the sort order.
9. Choose **Create Data View** to start the view creation process.

The new view is listed in the **Data Views** table. The details of the view are shown in the **Details** link under **View Last Updated** column.

> **Note**
>
> For small files of up to 100 megabytes, data view creation takes approximately 2 minutes. For larger files of around 1 gigabyte, expect data view creation to take approximately 3-4 minutes. Views with partitioning and sorting schemes may take longer.

10 If you chose to access the data view in a FinSpace notebook using Spark, the **Analyze in Notebook** button appears on the right side of the view. Choose this button to open the data view in a FinSpace notebook.

If you chose to access the data view using the FinSpace API, an **External API Access** button appears on the right side of the view. Choose this button to open the **External Data View Access** dialog box.




The dialog box displays the following information that you can copy to access the external data view:

- **S3 path** – The location where the external data view is stored. This location is unique for every data view.
- **Details** – Format options selected at the time of creating the data view.
- **API Credentials** – The credentials required to access the external data view from the S3 location. These credentials are only valid for 60 minutes. After the credentials expire, you need to choose the refresh icon to generate new credentials.

# Sharing data views in Amazon FinSpace

**Important**

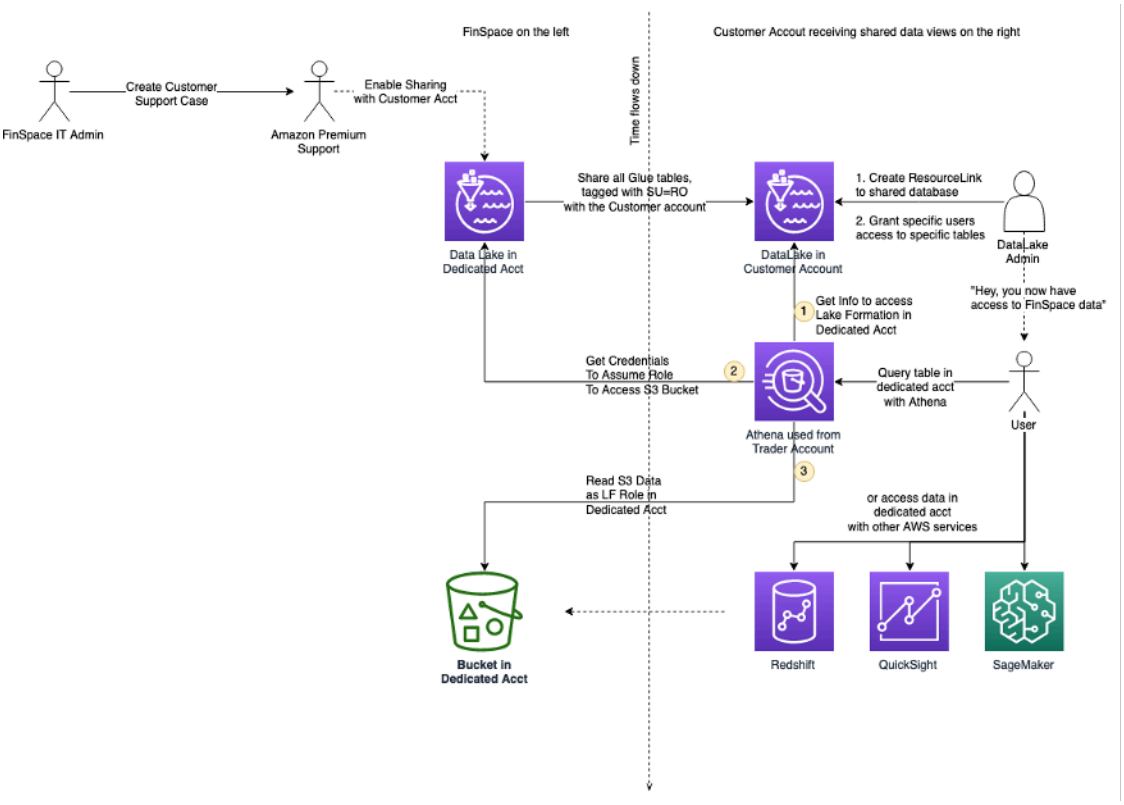
Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace stores your data in an AWS account called the *FinSpace environment infrastructure account*, which is a managed AWS account that's dedicated to your FinSpace environment. This account is separate from the account that you create your FinSpace environment in.

Data that you ingest into FinSpace is stored in the infrastructure account. You can access this data through data views. Data views store a copy of your data, which is organized for querying through an interface that is compatible with AWS Glue tables. You can query this interface by using the managed Apache Spark clusters in FinSpace.

With FinSpace data view sharing, you can share these tables with a Lake Formation data lake. When you do this, you can easily query the data with AWS analytics engines like Amazon Redshift, Athena, Amazon QuickSight, Amazon EMR, and SageMaker.

The following diagram illustrates how you can access FinSpace data views with AWS integrated services.




- The diagram shows the first part of the process where a FinSpace IT admin creates a technical support case, to request enabling the FinSpace infrastructure account for sharing. The request consists of the identifier of the environment to be shared and the AWS Region.
- Next, the AWS support engineer enables the database and the data view tables to be shared in the designated FinSpace environment within the customer's account.
- A Lake Formation administrator in the customer's account creates a resource link to the shared database. Then, the administrator grants access to the resource link, the shared database, and the shared tables to other principals in the customer account.
- Finally, principals in the customer's account are able to access the FinSpace data view tables with AWS integrated services such as Athena, Amazon Redshift, Amazon QuickSight, and SageMaker.

## Tutorial: Sharing data views using AWS Lake Formation

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be

affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

This tutorial guides you through steps to enable access to data views and query the ingested data by using the integrated services. The topics in this tutorial explain how to create a resource link and use it to anable access to the data views in the infrastructure account.

**Prerequisites and considerations**

Before you start this tutorial, complete the following prerequisites:

- Make sure that your Amazon FinSpace environment is enabled to share data views. You can request that your environment is enabled by creating a technical support case at AWS Support. For this, choose the service as **Amazon FinSpace**.

  In the support case, specify that you want to enable data view sharing for your FinSpace environment. Make sure that you include the environment ID and the Region when you create the support case. For more information, see Creating support cases in the *AWS Support User Guide*.

  After data sharing is enabled in your FinSpace environment, all the data views of the internal datasets in FinSpace are instantly available in the target Lake Formation catalog as a Lake Formation table.

  **Note**

  - If you have want to share data views from multiple environments, you need to create a separate support case for each environment.
  - If you want to disable data view sharing for your FinSpace environment, you need to create a new technical support case.

- Make sure that users or roles in the customer account have access to use Lake Formation and other required analytics engines such as Amazon Redshift, Athena, Amazon QuickSight, Amazon EMR, and SageMaker.
- If you want to request data from an integrated service like Amazon Athena, ensure that an Amazon S3 location is configured.
- At least one user must be a Lake Formation data lake administrator to view shared resources.

### Step 1: Enable the link to the shared database

To enable access to the shared data view, first you need to create a resource link. A resource link is a Data Catalog object that is a link to a shared database or table.

After you create a resource link and grant data permissions, you can use integrated services to run queries on the shared databases or tables. For more information on resource links, see the AWS Lake Formation Developer Guide.

**To create a resource link to a shared database**

1. Open the AWS Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. In the navigation pane, choose **Databases**, and then choose the required shared database from the list. The shared database name starts with *finspace_*, followed by the Environment ID that you provided in the technical support case.

   > **Note**
   >
   > The shared database is not available in the list on this page unless you request access to it by creating a support case.

3. Choose **Actions** and then choose **Create resource link**.
4. On the **Create resource link** page, enter the resource link name. For the subsequent query, it's helpful if the resource link name is the same as the database name.
5. Choose **Create**. The resource link is created, and you can view the resource link name in italics under the **Name** column on the **Databases** page.

After you create a resource link, only you can view and access it. To allow other principals in your account to access the resource link, grant permissions on the resource link.

**To grant data permissions on the resource link**

1. On the **Databases** page, under the **Name** column, choose the resource link name in italics.
2. Choose **Actions** and then choose **Grant**.
3. On the **Grant data permissions** page, under **Principals**, choose **IAM users and roles**.
4. Choose one or more users or roles from the **IAM users and roles** list.

5. In the **LF-Tags or catalog resources** section, choose **Named data catalog resources**, and then choose one or more databases to grant permissions to.
6. For the **Resource link permissions** section, choose *Describe*.
7. Choose **Grant**. You can now use the resource link to access the shared database.

> **Note**
>
> Granting permissions on a resource link doesn't grant permissions on the target (shared) database or table. You must grant permissions on the target database separately.

### Step 2: Enable access to the target database and tables

After creating the resource link with permissions, you can use it to grant access to the shared database and tables.

**To grant access to the shared database**

1. Open the AWS Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. In the navigation pane, choose **Databases**. Then, from the list, choose the resource link that you created in the previous step.
3. Choose **Actions** and then choose **Grant on target**.
4. On the **Grant data permissions** page, under **Principals**, choose **IAM users and roles**.
5. Choose one or more users or roles from the **IAM users and roles** list.
6. In the **LF-Tags or catalog resources** section, choose **Named data catalog resources**. Then, from the list, add the database that you selected while granting access to the resource link.
7. For the database permissions , select *Describe*.
8. Choose **Grant**.

**To grant access to the tables**

1. Open the AWS Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. In the navigation pane, choose **Tables**, and then choose the required table from the list.
3. Choose **Actions** and then choose **Grant**.

4. On the **Grant data permissions** page, under **Principals**, choose **IAM users and roles**.
5. Choose one or more users or roles from the **IAM users and roles** list.
6. In the **LF-Tags or catalog resources** section, choose **Named data catalog resources**. Then, from the list, choose the table that you want to grant permissions to.
7. For the table permissions, select *Describe* and *Select*.
8. Choose **Grant**.

### Step 3: Query data by using integrated services

After you have access to the database and tables, you can use the integrated services to query data in the tables.

The following procedure shows how to query data using Amazon Athena.

> **Note**
>
> Before you can run a query using Athena, you must specify a query result bucket in Amazon S3. To set up an Amazon S3 query result location, see Specifying a query result location in the *Amazon Athena User Guide*.

### To query data using Amazon Athena

1. Open the AWS Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. In the navigation pane, choose **Tables**, and then choose the required table name from the list.
3. On the **Table details** page, choose **Actions** and then choose **View data**. You are taken to Athena to preview data.
4. Choose **Ok** in the dialog box to confirm navigating to Amazon Athena. The Athena query editor opens in a new browser tab.

   > **Note**
   >
   > You will be charged separately for Athena queries.

5. In the Athena query editor, enter the SQL query and choose **Run**.

   You can view the query results at the bottom of the page.

> **Note**
>
> You must specify the resource link to include shared resources in the query.

You can also use Amazon Redshift to query data in the data lake. For more information, see the AWS Lake Formation Developer Guide.

# Prepare and analyze data in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can perform data preparation and analysis on datasets in Amazon FinSpace. You cannot analyze datasets directly, instead you can create data views from datasets that enable you to perform analysis.

**Topics**

- Working with Amazon FinSpace notebooks
- Working with Spark clusters in Amazon FinSpace
- Importing library in Amazon FinSpace
- Accessing Amazon S3 Bucket from FinSpace notebook
- Amazon FinSpace Spark time series library

# Working with Amazon FinSpace notebooks

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace notebook provides an Integrated Development Environment (IDE) that lets you access data from the FinSpace Data Catalog to perform data preparation and analysis. FinSpace simplifies the use of Apache Spark providing access to fully managed Spark Clusters using easy to launch cluster templates. For more information, see Apache Spark.

**Note**

- In order to use notebooks and Spark clusters, you must be a superuser or a member of a group with necessary permissions - **Access Notebooks, Manage Clusters**.
- The Spark clusters are terminated daily at midnight US Eastern time.

FinSpace notebooks are programmed using Python. Python and Spark integration is achieved using the PySpark library. For more information, see PySpark.

**Topics**

# Opening the notebook environment

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

**Note**

- In order to open a notebook environment, you must be a superuser or a member of a group with necessary permissions - **Access Notebooks**.
- Expect a one-time setup delay of 15-20 minutes for the notebook environment after creating a new user.

**You can open a notebook environment in the following ways**

- Using the data view cards on homepage.
- From the dataset details page under **Data Overview** tab.
- From the dataset details page under **All Data Views** tab.

**Access notebook from homepage**

**To access notebook environment from the recently created data views**

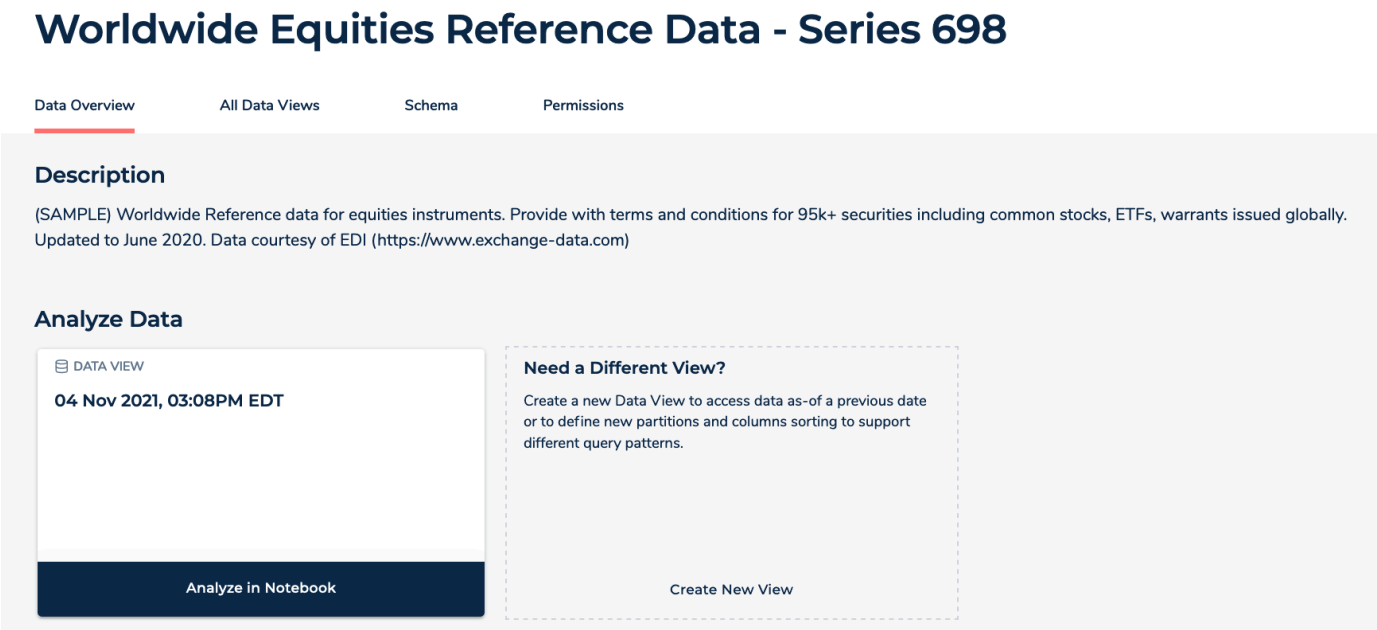
1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. From the homepage, under the **Status of Data Views** section, find the recently created data view.
3. On the data view card, choose **Analyze**. The notebook opens in a new tab on your browser.

### Access notebook from Data Overview tab

#### To access notebook environment from the overview tab

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. From the homepage, search for a dataset.
3. Choose the dataset name to view the dataset details page.
4. From the **Data Overview** tab, under **Analyze Data** section, choose **Analyze in Notebook** in the data view card.

   The notebook opens in a new tab on your browser.



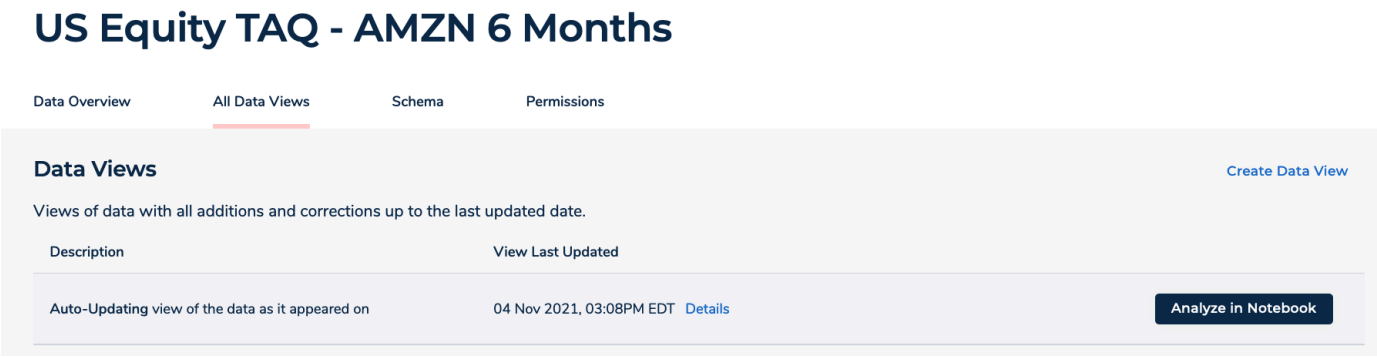

### Access notebooks from All Data Views tab

#### To access notebook environment from the list of all data views for a dataset

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. From the homepage, search for a dataset.
3. Choose the dataset name to view the dataset details page.
4. Choose **All Data Views** tab.

5. From the **Data Views** table, choose **Analyze in Notebook** for any of the data views.

   The notebook opens in a new tab on your browser.




## Working in the notebook environment

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Choosing **Go to Notebook** or **Analyze in Notebook** will open Jupyter Lab in a new tab in your web browser. You will land in the launcher page of SageMaker studio.

### To start a notebook with FinSpace kernel

1. In the upper-left corner of SageMaker Studio, choose **Amazon SageMaker Studio** to open Studio Launcher.
2. On the **Launcher** page, choose **Notebooks and compute resources**.
3. For **Select a SageMaker image**, choose the FinSpace PySpark image.
4. Choose **Notebook** to create a notebook in the FinSpace PySpark image.

### FinSpace kernel

The FinSpace PySpark Kernel comes with all the libraries required to access and work with data stored in FinSpace, including the Spark Cluster management API and time series analytics library. The FinSpace Cluster Management API is used to instantiate and connect the notebook instance to a dedicated Spark Cluster. FinSpace Spark clusters use Kerberos authentication for additional security. FinSpace provides with complete resource isolation when working with Spark Clusters.

When a FinSpace PySpark Kernel is instantiated for the first time in a new notebook session, you can expect a startup time of about 3 to 5 minutes to allow bootstrapping of all dependencies on the image supporting the notebook.

## Access datasets from a notebook

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can conveniently and securely access all datasets to prepare and analyze data from your Amazon FinSpace notebook. The following sections show how to access data from a FinSpace notebook.

**Note**

In order to use notebooks and Spark clusters, you must be a superuser or a member of a group with necessary permissions - **Access Notebooks, Manage Clusters**.

### Access data using a pre-populated notebook

### To access data using a pre-populated notebook

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.

2. Open a notebook by using one of the three methods listed in Opening the notebook environment.

   In the notebook, the dataset ID and data view ID are pre-populated.

3. Run all cells to print the schema and content of the data view.

### Access data using a newly created notebook

#### To access data using a newly created notebook

1. Run the following code from your notebook to instantiate a cluster and connect the FinSpace PySpark image to the cluster.

```
%local
from aws.finspace.cluster import FinSpaceClusterManager

finspace_clusters = FinSpaceClusterManager()
finspace_clusters.auto_connect()
```

   The output should be similar to the following output

```
Cluster is starting. It will be operational in approximately 5 to 8 minutes
Started cluster with cluster ID: 8x6zd9cq and state: STARTING
......

cleared existing credential location
Persisted krb5.conf secret to /etc/krb5.conf
re-establishing connection...
Persisted keytab secret to /home/sagemaker-user/livy.keytab
Authenticated to Spark cluster
Persisted Sparkmagic config to /home/sagemaker-user/.Sparkmagic/config.json
Started Spark cluster with clusterId: 8x6zd9cq
finished reloading all magics & configurations
Persisted FinSpace cluster connection info to /home/sagemaker-user/.Sparkmagic/
FinSpace_connection_info.json

SageMaker Studio Environment is now connected to your FinSpace Cluster: 8x6zd9cq at
 GMT: 2021-01-15 02:13:50.
```

> **Note**
>
> Without the `%local` at the beginning of the cell, your code will be executed on the Spark cluster.

2. To access the data view, you will need the dataset ID and data view ID. To get these IDs

   a. In the FinSpace web application, open the dataset details page of the dataset that you want to analyze.

   b. Under the **All Data Views** tab, find the data view that you want to analyze.

   c. Choose **Details**.

   d. Copy the **Data View ID** and **Dataset ID** to use in the notebook.

3. Initialize dataset ID and data view ID in the notebook.

```
dataset_id    = "rgg1hj1"
data_view_id  = "VrvKEKnA1El2nr821BaLTQ"
```

4. Instantiate FinSpace Analytics Manager to access the data and read into a Spark DataFrame.

```
from aws.finspace.analytics import FinSpaceAnalyticsManager
finspace_analytics = FinSpaceAnalyticsManager(Spark = Spark)

df = finspace_analytics.read_data_view(dataset_id = dataset_id, data_view_id =
 data_view_id)
```

## Example notebooks

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can access example notebooks and Python scripts illustrating how to use Amazon FinSpace to prepare and analyze data using Spark Clusters and the time series analytics library. Examples notebooks are available on github. You can clone the gitrepo in your Jupyter Lab for easy access to the example notebooks.

## Working with Spark clusters in Amazon FinSpace

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace simplifies how to work with Spark clusters by offering easy to use cluster configuration templates that allow you to launch, connect, resize, and terminate without worrying to manage the underlying infrastructure. Every user in FinSpace with **Access Notebooks** and **Manage Clusters** permission can instantiate one cluster.

**Note**

In order to use notebooks and Spark clusters, you must be a superuser or a member of a group with necessary permissions - **Access Notebooks, Manage Clusters**.

You can choose one of the following cluster configuration templates:

- Small
- Medium
- Large
- XLarge
- 2XLarge

> **Note**
>
> You are charged by the minute for using the Spark clusters. Terminate your Spark cluster when you are done using it.

## Import FinSpace cluster management library

Use the following code to import the cluster management library in a notebook.

```
%local
from aws.finspace.cluster import FinSpaceClusterManager
```

## Start a Spark cluster

Use the following code to start and connect your notebook to a Spark cluster.

```
%local
from aws.finspace.cluster import FinSpaceClusterManager

finspace_clusters = FinSpaceClusterManager()
finspace_clusters.auto_connect()
```

For a newly created cluster, the output should be similar to the following.

```
Cluster is starting. It will be operational in approximately 5 to 8 minutes
Started cluster with cluster ID: 8x6zd9cq and state: STARTING
......

cleared existing credential location
Persisted krb5.conf secret to /etc/krb5.conf
re-establishing connection...
Persisted keytab secret to /home/sagemaker-user/livy.keytab
Authenticated to Spark cluster
Persisted Sparkmagic config to /home/sagemaker-user/.Sparkmagic/config.json
Started Spark cluster with clusterId: 8x6zd9cq
finished reloading all magics & configurations
Persisted FinSpace cluster connection info to /home/sagemaker-user/.Sparkmagic/
FinSpace_connection_info.json
```

```
SageMaker Studio Environment is now connected to your FinSpace Cluster: 8x6zd9cq at
 GMT: 2021-01-15 02:13:50.
```

You can expect a startup time of about 5 to 8 minutes when instantiating a cluster for the first time. Once a cluster is created, any newly created notebook will detect and connect to the running cluster when an `auto_connect()` call is issued and this operation is instantaneous.

## List details for Spark clusters

### Use the following code to list the Spark cluster name and details

```
%local
finspace_clusters.list()
```

The output should be similar to the following output.

```
{'clusters': [{'clusterId': '8x6zd9cq',
   'clusterStatus': {'state': 'RUNNING',
    'reason': 'Started successfully',
    'details': ''},
   'name': 'hab-cluster-3e51',
   'currentTemplate': 'FinSpace-Small',
   'requestedTemplate': 'FinSpace-Small',
   'clusterTerminationTime': 1610676314,
   'createdTimestamp': 1610676374420,
   'modifiedTimestamp': 1610676823805},
  {'clusterId': '3ysaqx3g',
   'clusterStatus': {'state': 'TERMINATED',
    'reason': 'Initiated by user',
    'details': ''},
   'name': 'hab-cluster-c4f9',
   'currentTemplate': 'FinSpace-Small',
   'requestedTemplate': 'FinSpace-Small',
   'clusterTerminationTime': 1610478542,
   'createdTimestamp': 1610478602457,
   'modifiedTimestamp': 1610514182552}]}
```

In the output above, you can see the `clusterId` **8x6zd9cq** is a small cluster with state equals to **RUNNING**, and the `clusterId` **3ysaqx3g** is a small cluster with state equals to **TERMINATED**.

## Resize Spark cluster

Scale your Spark cluster up or down based on your compute needs and the volume of data you need to analyze.

**To resize clusters**

1. Type the following code to update your cluster to a **Large** size.

```
%local
finspace_clusters.update('8x6zd9cq','Large')
```

   The output will look like below

```
{'clusterId': '8x6zd9cq',
 'clusterStatus': {'state': 'UPDATING', 'reason': 'Initiated by user'}}
```

2. Note that the `update()` operation runs asynchronous so that you can continue to work on the cluster as the update operation completes.
3. Check the status of the update operation using the `list()` function.

```
{'clusters': [{'clusterId': '8x6zd9cq',
   'clusterStatus': {'state': 'UPDATING',
    'reason': 'Initiated by user',
    'details': ''},
   'name': 'hab-cluster-3e51',
   'currentTemplate': 'Small',
   'requestedTemplate': 'Large',
   'clusterTerminationTime': 1610676314,
   'createdTimestamp': 1610676374420,
   'modifiedTimestamp': 1610682765327}}
```

4. In the output above, the `clusterId` **8x6zd9cq** is being updated from **Small** to **Large**.

## Terminate Spark cluster

Terminate your Spark cluster once your work is done, so that you don't incur additional charges.

**To terminate your Spark cluster**

1. Type the following code to terminate a cluster.

```
%local
finspace_clusters.terminate('8x6zd9cq')
```

2. You can check the state of the cluster using the `list()` function.

## Importing library in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can install notebook-scoped libraries on a running Amazon FinSpace cluster directly via a FinSpace notebook. This capability is useful in scenarios in which you do not have access to a PyPI repository but need to analyze and visualize a dataset.

Notebook-scoped libraries provide you the following benefits:

- **Runtime installation** – You can import Python libraries from PyPI repositories and install them on your remote cluster on the fly when you need them. These libraries are instantly available to your Spark runtime environment. There is no need to restart the notebook session or recreate your cluster.
- **Dependency isolation** – The libraries you install using FinSpace notebooks are isolated to your notebook session and don't interfere with bootstrapped cluster libraries or libraries installed from other notebook sessions. These notebook-scoped libraries take precedence over bootstrapped libraries. Multiple notebook users can import their preferred version of the library and use it without dependency clashes on the same cluster.
- **Portable library environment** – The library package installation happens from your notebook file. This allows you to recreate the library environment when you switch the notebook to a different cluster by re-executing the notebook code. At the end of the notebook session, the libraries you install through FinSpace notebooks are automatically removed from the hosting cluster.

The following example code shows how to install pandas and matplotlib from the PiPY repository.

```
sc.install_pypi_package("pandas==0.25.1") #Install pandas version 0.25.1
sc.install_pypi_package("matplotlib", "https://pypi.org/simple") #Install matplotlib
 from given PyPI repository
```

You can uninstall packages using the `uninstall_package` PySpark API.

```
sc.uninstall_package('pandas')
```

## Accessing Amazon S3 Bucket from FinSpace notebook

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

An Amazon FinSpace environment can be configured to access your Amazon S3 buckets from FinSpace notebook.

**Note**

In order to setup access to an S3 bucket, you must be authorized to access the FinSpace page in AWS Management Console and make changes to bucket-level permissions in Amazon S3.

**To find your infrastructure account number**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. On the FinSpace console, from the list of environments, choose the environment that you want to setup to access an S3 bucket. If there are no environments available, create one by following the steps listed in Create an Amazon FinSpace environment.

3. On the environment page, copy and save the FinSpace infrastructure account name.

**To setup access for FinSpace infrastructure account in S3 bucket policy**

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Choose the bucket that you want to access from your FinSpace environment.
3. Set a bucket policy for the bucket with following json code. For example, if your bucket name is `example-bucket` and your FinSpace infrastructure account number is `123456789101` below would be the example policy.

```
{
    "Version": "2012-10-17",
    "Id": "CrossAccountAccess",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789101:role/FinSpaceServiceRole"
                ]
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::example-bucket/*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789101:role/FinSpaceServiceRole"
                ]
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::example-bucket"
        }
    ]
}
```

Using the above policy, you should be able to access `example-bucket` from the Jupyter notebook of a FinSpace environment, which is associated with the FinSpace infrastructure account number `123456789101`.

## Amazon FinSpace Spark time series library

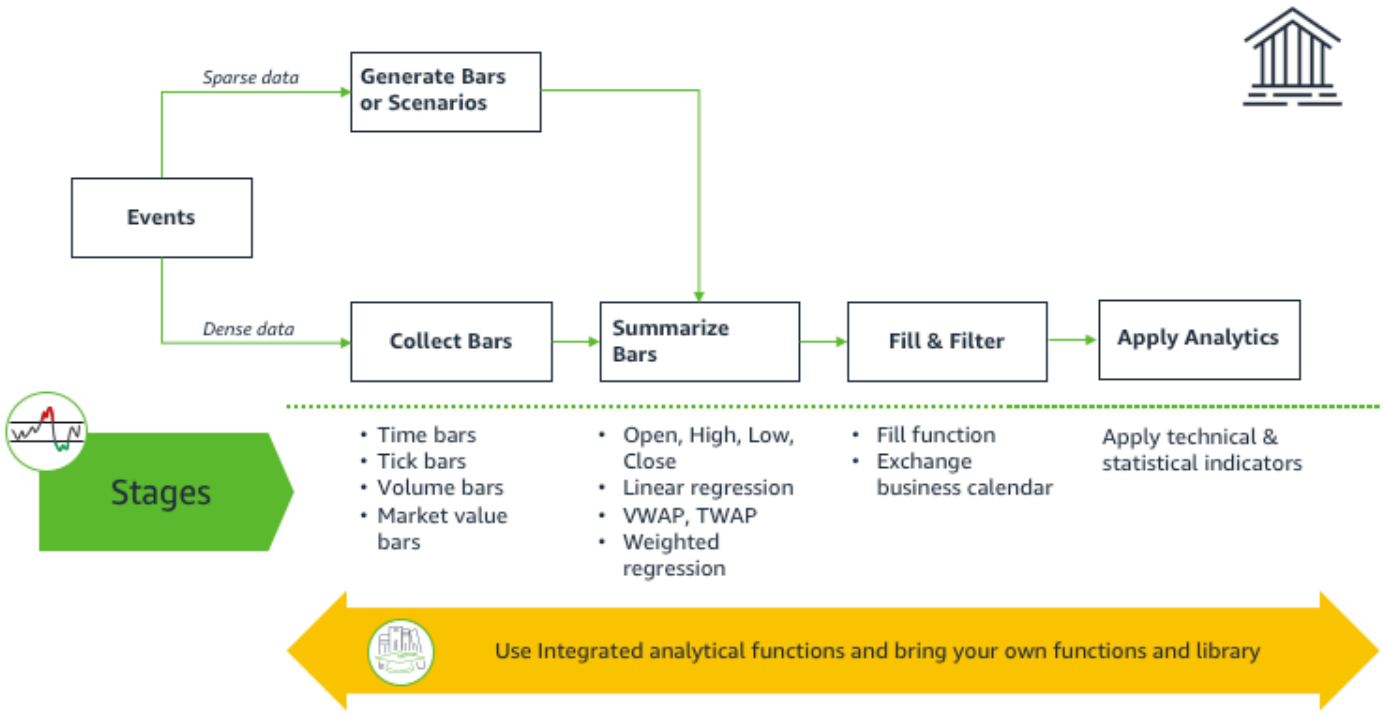
> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace PySpark Kernel delivers a time series analytics library to prepare and analyze historical financial time series data using FinSpace managed Spark clusters. You can use the time series library to analyze high-density data like US options historical Options Price Reporting Authority (OPRA) with billions of daily events or sparse time series data such as quotes for fixed income instruments. The time series library is available to use in the FinSpace notebook environment.

The time-series library is logically organized in four stages for a conceptual framework. Every stage provides a set of functions and you can plug your own functions.

1. **Collect** – The objective of this stage is to collect the series of events that arrive at an irregular frequency into uniform intervals called bars. You can perform collection with your functions or use the FinSpace functions to calculate bars such as time bars.
2. **Summarize** – The objective of this stage is to take collected data in bars from previous stage and summarize it using the events captures within a bar.
3. **Fill and Filter** – The data produced in the previous stage could have missing bars where no data was collected or contain data that is not desired to be used in the next stage. The objective of this stage is to prepare a dataset of features with evenly spaced intervals and filter out any data outside desired time window.

4. **Analytics** – At this stage, a prepared dataset of features is ready for application of technical and statistical indicators. You can bring your own indicator functions or choose one of the FinSpace functions for this stage.




See the following sections to learn more about supported functions in the time series library.

**Topics**

- Collect time bars operations in Amazon FinSpace
- Summarize bars operations in Amazon FinSpace
- Fill and filter operations in Amazon FinSpace
- Analyze operations in Amazon FinSpace
- Using the Amazon FinSpace library

## Collect time bars operations in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The objective of functions at this stage is to collect the series of events that arrive at an irregular frequency into uniform intervals called bars. You can perform collection with your functions or use the Amazon FinSpace functions to calculate bars. Collect functions are available in the `aws.finspace.timeseries.spark.windows` module and include the following list of functions.

### Compute analytics on features

```
aws.finspace.timeseries.spark.windows.compute_analytics_on_features(data, new_column,
 func, partition_col_list=None, add_intermediate=False)
```

Appends to data Dataframe, a new column whose value is computed by executing pandas user defined function (UDF) on a window of rows as specified by the function window dependency member.

### Parameters

- `data` (DataFrame) – input dataframe
- `new_column` (str) – name of new column to add
- `input_spec` – input specification
- `func` (Callable[…, Column]) – function to calculate over data
- `grouping_col_list` – a single or list of columns to group window on
- `add_intermediate` (Optional[bool]) – include intermediate data used in the calculation

**Return type** `DataFrame`

**Returns**

**Compute features on time bars**

```
aws.finspace.timeseries.spark.windows.compute_features_on_time_bars(data, new_column,
 func, force_ordering=False,*ordering_cols)
```

Reduces data by applying function preserving all other columns.

**Parameters**

- `data` (DataFrame) – input DataFrame
- `new_column` (str) – new column name
- `func` (Callable[…, Column]) – function to calculate over data
- `force_ordering` (Optional[bool]) – return data in sort in timecolumn order
- `ordering_cols` (str) – list of cols to orderBy on

**Return type** `DataFrame`

**Returns** `DataFrame`

**Create time bars**

```
aws.finspace.timeseries.spark.windows.create_time_bars(data, timebar_column,
 grouping_col_list, input_spec, timebar_spec, force_ordering=False)
```

Appends a column to the data frame in data with a rolling window of data. An optional `force_ordering` flag ensures that the rolling data is order by the `timebar_column`.

**Parameters**

- `data` (Union[Column, DataFrame]) – input dataframe
- `timebar_column` (str) – new timebar column name
- `grouping_col_list` (Union[str, List[str]]) – list of columns to group results on
- `input_spec` (BarInputSpec) – the input spec used to generate the time bars
- `timebar_spec` (Union[TimeBarSpec, Column]) – the timebar spec used to generate the time bars

- `force_ordering` (Optional[bool]) – optional force ordering in windows

**Return type** `DataFrame`

**Returns** `DataFrame`

**Spark spec module**

**Bar input spec**

```
class aws.finspace.timeseries.spark.spec.BarInputSpec(bar_structure_name,
 *bar_value_columns)
```

Bases: object

This class is responsible for modeling the input specification of bar operations.

**Calc input spec**

```
class aws.finspace.timeseries.spark.spec.CalcInputSpec(timestamp_column,
 holiday_calendar=<aws.finspace.finance.calendars.USEndOfDayCalenobject>,
 **kwargs_func_to_column)`
```

Bases: object

This class is responsible for modeling the input specification of calculation operations.

**Time bar spec**

```
class aws.finspace.timeseries.spark.spec.TimeBarSpec(timestamp_column, window_duration,
 slide_duration=None, start_time=None)
```

Bases: object

This class models the input time window specification, and associated calendar.

**to_window()**

Create an equivalent spark window from TimeBarSpec.

## Summarize bars operations in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The objective of this stage is to take collected data in bars from previous stage and summarize it using the events captures within a bar. Collect functions are available in the `aws.finspace.timeseries.spark.summarizer` module and include the following list of functions.

**Bar count**

```
aws.finspace.timeseries.spark.summarizer.bar_cnt(input_series)
```

**Returns** the number of items in interval.

**Parameters**

- `input_series` (Series) – a series window produced through groupby

**Return type** `Series`

**Returns** `pandas.Series`

**Close**

```
aws.finspace.timeseries.spark.summarizer.close(price)
```

Returns the last row, called close as its the closing price of that interval.

**Parameters**

- `price` (Series) – a series Window produced through group by

**Return type** `Series` **Returns** `pandas.Series`

### First last high low

```
aws.finspace.timeseries.spark.summarizer.first_last_high_low(sort_col: list, price:
 list) -> list
```

**Return type** `list`

### First last high low presorted

```
aws.finspace.timeseries.spark.summarizer.*first_last_high_low_presorted*(price: list) -
> list
```

**Return type** `list`

### High

```
aws.finspace.timeseries.spark.summarizer.high(price)
```

Returns the highest price in that interval.

**Parameters**

- `price` (Series) – a DataFrame Window produced through groupby

**Return type** `Series`

**Returns** `pandas.Series`

### Low

```
aws.finspace.timeseries.spark.summarizer.low(price)
```

Returns the lowest price in that interval.

**Parameters**

- `price` (Series) – a series Window produced through groupby

**Return type** `Series`

**Returns** `pandas.Series`

### Low high

```
aws.finspace.timeseries.spark.summarizer.lowhigh(value) -> list
```

**Return type** `list`

### Open high low close (OHLC)

The first, high, low, and last value over an interval.

```
aws.finspace.timeseries.spark.summarizer.ohlc_func(sort_col: list, price: list) -> list
```

**Return type** `list`

### Open high low close pre-sorted (OHLC)

The first, high, low, and last value over an interval.

```
aws.finspace.timeseries.spark.summarizer.ohlc_func_pre_sorted(price: list) -> list
```

**Return type** `list`

### Open high low close scala (OHLC)

The first, high, low, and last value over an interval.

```
aws.finspace.timeseries.spark.summarizer.ohlc_scala(timeseries, values)
```

### Open

```
aws.finspace.timeseries.spark.summarizer.open(price)
```

Returns the first row, the opening price over that interval.

**Parameters**

- `price` (Series) – a series window produced through groupby

**Return type** `Series` **Returns** `pandas.Series`

**Standard deviation**

```
aws.finspace.timeseries.spark.summarizer.std(price)
```

Returns the standard deviation over that interval.

**Parameters**

- `price` (Series) – a series Window produced through groupby

**Return type** `Series` **Returns** `pandas.Series`

**Time Delta**

```
aws.finspace.timeseries.spark.summarizer.*time_delta*(time_series: list, ref_date:
 datetime.date) -> list
```

**Return type** `list`

**Total volume**

```
aws.finspace.timeseries.spark.summarizer.total_volume(volume)
```

The total volume over that interval.

**Parameters**

- `volume` (Series) – input volume

**Return type** `DataFrame`

**Returns**

**Volume and close**

```
aws.finspace.timeseries.spark.summarizer.volume_and_close(price: list, vol: list) ->
 list
```

**Return type** `list`

### Volume weighted average price (VWAP)

```
aws.finspace.timeseries.spark.summarizer.vwap(price, volume)
```

The volume weighted average price over that interval.

**Parameters**

- `price` (Series) – input price series
- `volume` (Series) – input volume

**Return type** `DataFrame`

**Returns**

## Fill and filter operations in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The data produced after summarizing bars could have missing bars where no data was collected or contain data that is not desired to be used in the next stage. The objective of this stage is to prepare a dataset with evenly spaced intervals and filter out any data outside desired time window. Fill and Filter are available in the `aws.finspace.timeseries.spark.prepare` module.

### Fill and filter functions

```
aws.finspace.timeseries.spark.prepare.time_bar_fill_and_filter(data,
timebar_column_name, business_calendar, time-bar_spec, start_date,
  end_date, fill_value=None, start_time=None, end_time=None)
```

The data produced after summarizing bars could have missing bars where no data was collected or contain data that is not desired to be used in the next stage. The objective of this stage is to prepare a dataset with evenly spaced intervals and filter out any data outside desired time window. Fill and Filter are available in the `aws.finspace.timeseries.spark.prepare` module.

The Fill and filter function will fill with nulls in all rows that need to exist, and filter all rows that are outside the business calendar date/time range in a given calendar.

**Parameters**

- `data` (DataFrame) – input dataframe
- `timebar_column_name` (str) – name of the timebar column to fill against
- `business_calendar` (AbstractCalendar) – business calendar
- `timebar_spec` (TimeBarSpec) – time bar input spec associated with the bars that were created. it provides the bar frequency
- `start_date` (date) – start of date
- `end_date` (date) – end date
- `fill_value` (Optional[float]) – value to fill
- `start_time` (Optional[time]) – start time of the day
- `end_time` (Optional[time]) – end time of the day

**Return type** `DataFrame`

**Returns** `DataFrame`

**Calendars module**

Use the calendar module for defining a calendar schedule to be used in fill and filter.

**Abstract calendar**

```
class aws.finspace.finance.calendars.AbstractCalendar
Bases: object
```

Defines abstract class for calendars.

```
DISRUPTIONS = 'DISRUPTIONS'
```

```
EARLY_CLOSINGS = 'EARLY_CLOSING'
EARLY_CLOSING_TIME = datetime.time(13, 30)
END_OF_TRADING = 'END_OF_TRADING'
HOLIDAYS = 'HOLIDAYS'
START_OF_TRADING = 'START_OF_TRADING'
TZINFO ='TZINFO'
```

### Create schedule

```
create_schedule_from_to(from_date, to_date, time_bar_spec_window_duration,
 from_time=None, to_time=None, tzinfo=<UTC>)
Abstract method, provide override
```

Creates a list of dates associated with a particular type of calendar.

### Parameters

- `from_date`(date) – from date
- `to_date` (date) – to date
- `time_bar_spec_window_duration` (str) –
- `from_time` (Optional[time]) – from time
- `to_time` (Optional[time]) – to time

**Return type** `array`

**Returns** raw_calendar_data()

Return type `Dict[str,Any]`

Returns raw calendar data

### NYSE calendar

```
class aws.finspace.finance.calendars.NYSECalendar20192020
Bases: aws.finspace.finance.calendars.USEndOfDayCalendarActAct_NoWeekends
```

Returns a holiday calendar with no weekends, and according to the NYSE exchange trading holidays and half-days for 2019 and 2020.

```
create_schedule_from_to(from_date, to_date, time_bar_spec_window_duration,
 from_time=None, to_time=None)
```

**Parameters**

- `from_date` (date) – from date
- `to_date` (date) – to date
- `time_bar_spec_window_duration` (str)
- `from_time` (Optional[time]) – from time
- `to_time` (Optional[time]) – to time
- `tzinfo` – time to localize to

**Return type** `array`

**Returns** raw_calendar_data()

Return type `Dict[str,Any]`

Returns raw calendar data

**End of day calendar actual**

```
class aws.finspace.finance.calendars.USEndOfDayCalendarActAct_NoWeekends
Bases: aws.finspace.finance.calendars.AbstractCalendar
```

Return 30/360 calendar, without weekends, without exchange hours.

```
create_schedule_from_to(from_date, to_date, time_bar_spec_window_duration,
 from_time=None, to_time=None)
```

**Parameters**

- `from_date` (date) – from date
- `to_date` (date) – to date
- `time_bar_spec_window_duration` (str)
- `from_time` (Optional[time]) – from time
- `to_time` (Optional[time]) – to time

- `tzinfo` – time to localize to

**Return type** `array`

**Returns** raw_calendar_data()

Return type `Dict[str,Any]`

Returns raw calendar data

## Analyze operations in Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

At this stage, a prepared dataset of features is ready for application of technical and statistical indicators. You can bring your own indicator functions or choose one of the FinSpace functions for this stage.

### Acceleration bands (ABANDS)

```
aws.finspace.timeseries.spark.analytics.abands(tenor, time_col_name, price_col_name,
 high_col_name, low_col_name)
```

The Acceleration Bands (ABANDS) created by Price Headley plots upper and lower envelope bands around a simple moving average.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

- high_col_name (str) – input array of high asset prices
- low_col_name (str) – input array of high asset prices

**Return type** Callable[. . . , Column]

**Returns** pandas/Spark user defined scalar function

**Accumulation/Distribution (AD)**

```
aws.finspace.timeseries.spark.analytics.acc_dist_indicator(time_col_name,price_col_name,
 high_col_name, low_col_name, volume_col_name)
```

The Accumulation/Distribution (AD) study attempts to quantify the amount of volume flowing into or out of an instrument by identifying the position of the close of the period in relation to that period's high/low range. The volume for the period is then allocated accordingly to a running continuous total. In this indicator, if the divisor, high-low is 0, and hence the current money flow volume is nan, then it means that price, which must fall between high and low is also going to equal high. In that case the numerator is 0 as well which means that the contribution should really be 0 in this case. Hence below we filter the NANs out in the equation. https://www.investopedia.com/terms/a/accumulationdistribution.asp

**Parameters**

- time_col_name (str) – name of time column
- price_col_name (str) – input array of asset prices, determined by user, default is close for intraday calculations
- high_col_name (str) – input array of high asset prices
- low_col_name (str) – input array of high asset price
- volume_col_name (str) – asset volume for the bar

**Return type** Callable[. . . , Column]

**Returns** pandas/Spark user defined scalar function

**Average directional movement index rating (ADXR)**

```
aws.finspace.timeseries.spark.analytics.adrx_indicator(adx_period, period,
 time_col_name, price_col_name, high_col_name, low_col_name)
```

The Average Directional Movement Index Rating (ADXR) is an element of the Directional Movement System, developed by J. Welles Wilder. ADXR quantifies the change in momentum of the Average Directional Index (ADX). This indicator is the result of adding two values of the Average Directional Index (the current ADX value and the ADX value n-periods ago), after which dividing this sum by two, or: `ADXR = (ADX + ADX n-periods ago) / 2`

**Parameters**

- `adx_period` (int) – look back window for Average Directional Index
- `period` (int) – look back window for Average Directional Index
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Average directional movement index (ADX)**

```
aws.finspace.timeseries.spark.analytics.adx_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

The Average Directional Movement Index (ADX) is designed to quantify trend strength by measuring the amount of price movement in a single direction. The ADX is part of the Directional Movement system published by J. Welles Wilder, and is the average resulting from the Directional Movement indicators.

**Parameters**

- `tenor` – window size
- `time_col_name` – name of time column
- `price_col_name` – input array of closing prices over bar
- `high_col_name` – input array of high prices over bar
- `low_col_name` – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Moving average convergence divergence (MACD)**

```
aws.finspace.timeseries.spark.analytics.apo_indicator(short_tenor, long_tenor,
 time_col_name, input_array_col_name)
```

The Moving Average Convergence Divergence (MACD) was developed by Gerald Appel, and is based on the differences between two moving averages of different lengths, a Fast and a Slow moving average. A second line, called the Signal line is plotted as a moving average of the MACD. A third line, called the MACD Histogram is optionally plotted as a histogram of the difference between the MACD and the Signal Line. Learn more.

**Parameters**

- `short_tenor` (int) – short window size
- `long_tenor` (int) – long window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Aroon down indicator**

```
aws.finspace.timeseries.spark.analytics.aroon_down_indicator(tenor, time_col_name,
 price_col_name)
```

`Aroon down indicator = ((Number of periods - Number of periods since lowest low) / Number of periods) * 100`

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column

- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Aroon oscillator**

```
aws.finspace.timeseries.spark.analytics.aroon_oscillator(tenor, time_col_name,
 price_col_name)
```

The Aroon Oscillator is a trend-following indicator that uses aspects of the Aroon Indicator (Aroon Up and Aroon Down)to gauge the strength of a current trend and the likelihood that it will continue. Readings above zero indicate that an uptrend is present, while readings below zero indicate that a downtrend is present. Traders watch for zero line crossovers to signal potential trend changes. They also watch for big moves, above 50 or below -50 to signal strong price moves.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Aroon up indicator**

```
aws.finspace.timeseries.spark.analytics.aroon_up_indicator(tenor, time_col_name,
 price_col_name)
```

`Aroon up indicator = ((Number of periods - Number of periods since highest high) / Number of periods) * 100`

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Average true range (ATR)**

```
aws.finspace.timeseries.spark.analytics.average_true_range(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

The Average True Range (ATR) study measures the size of the period's range, and takes into account any gap from the close of the previous period. Learn more. **Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Bollinger band (BBANDS)**

```
aws.finspace.timeseries.spark.analytics.bollinger_bands(tenor, no_std, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

The Bollinger Band (BBANDS) study created by John Bollinger plots upper and lower envelope bands around the price of the instrument. The width of the bands is based on the standard deviation of the closing prices from a moving average of price. Learn more.

**Parameters**

- `tenor` (int) – window to perform the calculation over
- **no_std** (int) – number of standard deviations
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** float

**Chaikin money flow**

```
aws.finspace.timeseries.spark.analytics.chaiken_money_flow_indicator(tenor,
 time_col_name, price_col_name, high_col_name, low_col_name, vol ume_col_name)
```

Developed by Marc Chaikin, Chaikin Money Flow measures the amount of Money Flow Volume over a specific period. Money Flow Volume forms the basis for the Accumulation Distribution Line. Instead of a cumulative total, Chaikin Money Flow sums Money Flow Volume for a specific look-back period, typically 20 or 21 days. The resulting indicator fluctuates above/below the zero line just like an oscillator. Chartists weigh the balance of buying or selling pressure with the absolute level of Chaikin Money Flow. Additionally, chartists can look for crosses above or below the zero line to identify changes on money flow.

**Parameters**

- `tenor` (int) – look back window
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar
- `volume_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Chaikens volatility indicator**

```
aws.finspace.timeseries.spark.analytics.chaikens_volatility_indicator(tenor,
 time_col_name, high_col_name, low_col_name)
```

Marc Chaikin's Volatility indicator compares the spread between a security's high and low prices, quantifying volatility as a widening of the range between the high and the low price.

**Parameters**

- `tenor` (int) – look back
- `time_col_name` (str) – name of time column
- `price_col_name` – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Chande momentum indicator**

```
aws.finspace.timeseries.spark.analytics.cmo_indicator(tenor, time_col_name,
 price_col_name)
```

The Chande Momentum Oscillator (CMO) developed by Tushar Chande attempts to capture the momentum of the instrument. The indicator oscillates between -100 and 100 with overbought level of 50 and oversold level of -50.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Commodity channel index (CCI)

```
aws.finspace.timeseries.spark.analytics.commodity_channel_index(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

The Commodity Channel Index (CCI) compares the current mean price with the average mean price over a typical window of 20 periods. Learn more.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Coppock curve

```
aws.finspace.timeseries.spark.analytics.coppock_curve_indicator(roc1_period,
 roc2_period, wma_period,  time_col_name, price_col_name)
```

The Coppock Curve is a long-term price momentum indicator used primarily to recognize major downturns and upturns in a stock market index. It is calculated as a 10-month weighted moving average of the sum of the 14-month rate of change and the 11-month rate of change for the index. It is also known as the Coppock Guide.

**Parameters**

- **roc1_period** (int) – rate of change 1 look back period
- **roc2_period** (int) – rate of change 2 look back period
- **wma_period** (int) – weighted moving average look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of high prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Debug UDF call**

```
aws.finspace.timeseries.spark.analytics.debug_udf_call(tenor, time_col_name, *kwargs)
```

**Return type** `Callable[. . . , Column]`

**Directional movement indicators (DMI)**

```
aws.finspace.timeseries.spark.analytics.dmi_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

The Directional Movement Indicators (DMI) are components of the Directional Movement system published by J. Welles Wilder, and are computed with the Average Directional Movement Index (ADX). Two indicators are plotted, a Positive DI ( +DI ) and a Negative DI ( -DI ).

**Parameters**

- `tenor` (int) – window size, typically 2 periods
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Donchian channels**

```
aws.finspace.timeseries.spark.analytics.donchian_channel_indicator(tenor,
 time_col_name, high_col_name, low_col_name)
```

Donchian Channels are three lines generated by moving average calculations that comprise an indicator formed by upper and lower bands around a mid-range or median band. The upper band marks the highest price of a security over N periods while the lower band marks the lowest price of

a security over N periods. The area between the upper and lower bands represents the Donchian Channel. Career futures trader Richard Donchian developed the indicator in the mid-twentieth century to help him identify trends. He would later be nicknamed **The Father of Trend Following**.

**Parameters**

- `tenor` *look back
- `time_col_name` *name of time column
- `high_col_name` *input array of high prices over bar
- `low_col_name` input array of low prices over bar

**Return type** *Callabl[. . . , Column]

**Return** pandas/spark user defined scalar function

**Double exponential moving average (DEMA)**

```
aws.finspace.timeseries.spark.analytics.double_exponential_moving_average(tenor,
 time_col_name, price_col_name)
```

The Double Exponential Moving Average (DEMA) by Patrick Mulloy attempts to offer a smoothed average with less lag than a straight exponential moving average. The calculation is more complex than just a moving average of a moving average as shown in the formula below.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]` **Returns** pandas/Spark user defined scalar function

**Detrended price oscillator (DPO)**

```
aws.finspace.timeseries.spark.analytics.dpo_indicator(tenor, time_col_name,
 price_col_name)
```

A detrended price oscillator is an oscillator that strips out price trends in an effort to estimate the length of price cycles from peak to peak or trough to trough. Unlike other oscillators, such as the stochastic or moving average convergence divergence (MACD), the DPO is not a momentum indicator. It highlights peaks and troughs in price, which are used to estimate buy and sell points in line with the historical cycle.

**Parameters**

- `tenor` – look back period
- `time_col_name` – name of time column
- `high_col_name` – input array of high prices over bar
- `low_col_name` – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Return** pandas/spark user defined scalar function

**Ease of movement indicator**

```
aws.finspace.timeseries.spark.analytics.ease_of_movement_indicator(tenor,
 time_col_name, high_col_name, low_col_name, vol- ume_col_name, scale=1000000)
```

Richard ArmsÃ¢ÂÂ Ease of Movement indicator is a technical study that attempts to quantify a mix of momentum and volume information into one value. The intent is to use this value to discern whether prices are able to rise, or fall, with little resistance in the directional movement. Theoretically, if prices move easily, they will continue to do so for a period of time that can be traded effectively.

**Parameters**

- `tenor` – look back window
- `time_col_name` – name of time column
- `high_col_name` – input array of high prices over bar
- `low_col_name` – input array of low prices over bar
- `volume_col_name` – input array of total volume over bar
- `scale` – scale multiplier

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Elder-ray index**

```
aws.finspace.timeseries.spark.analytics.elder_ray_index_indicator(tenor, time_col_name,
 price_col_name)
```

The Elder-Ray Index is a technical indicator developed by Dr. Alexander Elder that measures the amount of buying and selling pressure in a market. This indicator consists of three separate indicators known as **bull power** and **bear power**, which are derived from a 13-period exponential moving average (EMA). The three indicator help traders determine the trend direction and isolate spots to enter and exit trades.

**Parameters**

- `tenor` (int) – look back
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Exponential moving average**

```
aws.finspace.timeseries.spark.analytics.exponential_moving_average(tenor,
 time_col_name, input_array_col_name)
```

Compute exponential moving average on entire data set.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Stochastic fast (StochF)

```
aws.finspace.timeseries.spark.analytics.fast_stock_oscillator(tenor, time_col_name,
 price_col_name,  high_col_name, low_col_name)
```

The Stochastic Fast (StochF) normalizes price as a percentage between 0 and 100. Normally two lines are plotted, the %K line and a 3 day moving average of the %K which is called %D. A fast stochastic is created by not smoothing the %K line with a moving average before it is displayed.

### Parameters

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Fisher transform

```
aws.finspace.timeseries.spark.analytics.fisher_transformation_indicator(tenor,
 time_col_name, high_col_name, low_col_name)
```

The Fisher Transform is a technical indicator created by J.F. Ehlers that converts prices into a Gaussian normal distribution. In this way, the indicator highlights when prices have moved to an extreme, based on recent prices. This may help in spotting turning points in the price of an asset. It also helps show the trend and isolate the price waves within a trend.

### Parameters

- `time_col_name` (str) – name of time column

- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Force index**

```
aws.finspace.timeseries.spark.analytics.force_index_indicator(tenor, time_col_name,
 price_col_name, volume_col_name)
```

The force index is a technical indicator that measures the amount of power used to move the price of an asset. The term and its formula were developed by psychologist and trader AlexanderElder and published in his 1993 book Trading for a Living. The force index uses price and volume to determine the amount of strength behind a price move. The index is an oscillator, fluctuating between positive and negative territory. It is unbounded meaning the index can go up or down indefinitely.

**Parameters**

- `tenor` – look back window
- `time_col_name` – name of time column
- `price_col_name` – input array of closing prices over bar
- `volume_col_name` – input array of total volume over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Hull moving average (HMA)**

```
aws.finspace.timeseries.spark.analytics.hull_moving_average_indicator(tenor,
 time_col_name, price_col_name)
```

The Hull Moving Average (HMA) was developed by Alan Hull for the purpose of reducing lag, increasing responsiveness while at the same time eliminating noise. Its calculation is elaborate and

makes use of the Weighted Moving Average (WMA). It emphasizes recent prices over older ones, resulting in a fast-acting yet smooth moving average that can be used to identify the prevailing market trend.

**Parameters**

- `tenor` (int) – look back
- `time_col_name` (str) – name of time column
- `price_col_name` – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Ichimoku study**

```
aws.finspace.timeseries.spark.analytics.ichimoku_indicator(time_col_name,
 price_col_name, short_period=9, medium_period=26, long_period=52)
```

The Ichimoku study was developed by Goichi Hosoda pre-World War II as a forecasting model for financial markets. The study is a trend following indicator that identifies mid-points of historical highs and lows at different lengths of time and generates trading signals similar to that of moving averages/MACD. A key difference between Ichimoku and moving averages is Ichimoku charts lines are shifted forward in time creating wider support/resistance areas mitigating the risk of false breakouts. Learn more.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `short_period` (int) – short period window, usually 9
- `medium_period` (int) – long period window, usually 26
- `long_period` (int) – long period window, usually 52

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Kaufman adaptive moving average (KAMA)**

```
aws.finspace.timeseries.spark.analytics.kama_indicator(time_col_name, price_col_name,
 er_period=10, fast_ema_period=30, slow_ema_period=2)
```

Developed by Perry Kaufman, Kaufman's Adaptive Moving Average (KAMA) is a moving average designed to account for market noise or volatility. KAMA will closely follow prices when the price swings are relatively small and the noise is low. KAMA will adjust when the price swings widen and follow prices from a greater distance. This trend-following indicator can be used to identify the overall trend, time turning points and filter price movements.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices over bar
- **er_period** (int) – efficiency ratio tenor
- **slow_ema_period** (int)
- **fast_ema_period** (int)

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Keltner channel**

```
aws.finspace.timeseries.spark.analytics.keltner_indicator(time_col_name,
 price_col_name, high_col_name, low_col_name, atr_factor=2, ewm_tenor=20, atr_tenor=20)
```

The Keltner Channel was introduced in 1960 by Chester W. Keltner in his book **How To Make Money in Commodities**, and is also explained by Perry Kaufman's book **The New Commodity Trading Systems and Methods**. Keltner Channels plots three lines, consisting of a exponential moving average (typically of the average price) with upper and lower bands plotted above and below this moving average.The width of the bands is based on a user defined factor applied to the Average True Range, with this result added to and subtracted from the middle moving average line.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price
- **atr_factor** (float) – ATR multiplier
- **ewm_tenor** (int) – tenor of expo moving average
- **atr_tenor** (int) – tenor of ATR

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Klinger oscillator indicator**

```
aws.finspace.timeseries.spark.analytics.klinger_oscillator_indicator(short_period,
 long_period, time_col_name, price_col_name, high_col_name, low_col_name,
 volume_col_name)
```

The indicator was developed by Stephen Klinger to determine the long-term trend of money flow while remaining sensitive enough to detect short-term fluctuations. The indicator compares the volume flowing through securities with the security price movements and then converts the result into an oscillator. The Klinger oscillator shows the difference between two moving averages which are based on more than price. Traders watch for divergence on the indicator to signal potential price reversals. Like other oscillators, a signal line can be added to provide additional trade signals.

**Parameters**

- `short_period` (int) – short exponential moving average look back typically 34
- `long_period` (int) – long exponential moving average look back typically 55
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of high prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar
- `volume_col_name` (str) – input array of total volume over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Linear regression**

```
aws.finspace.timeseries.spark.analytics.linear_regression(tenor, time_col_name,
 input1_col_name, input2_col_name)
```

Takes two arrays of asset prices and then produces slope and intercept, where input1_col_name is the independent variable and input2_col_name is the dependent variable.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `input1_col_name` (str) – name of input array column
- `input2_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Linear weighted moving average**

```
aws.finspace.timeseries.spark.analytics.linear_weighted_moving_average(tenor,
 time_col_name, price_col_name)
```

Learn more.

**Parameters**

- `tenor` (int) – interval length
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Mass index

```
aws.finspace.timeseries.spark.analytics.mass_index_indicator(ema_period, sum_period,
 time_col_name, high_col_name, low_col_name)
```

Mass index is a form of technical analysis that examines the range between high and low stock prices over a period of time. Mass index, developed by Donald Dorsey in the early 1990s, suggests that a reversal of the current trend will likely take place when the range widens beyond a certain point and then contracts.

### Parameters

- `ema_period` – look back for exponential moving average typically 9 periods
- `sum_period` – look back for sum of exponential moving average typically 25 periods
- `time_col_name` – name of time column
- `high_col_name` – input array of high prices over bar
- `low_col_name` – input array of low prices over bar

**Return type** `Callable[. . . , Column]` **Returns** pandas/Spark user defined scalar function

### Max indicator

```
aws.finspace.timeseries.spark.analytics.max_indicator(tenor, time_col_name,
 price_col_name)
```

Compute max over look back period

### Parameters

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Money flow indicator

```
aws.finspace.timeseries.spark.analytics.mf_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name, vol ume_col_name)
```

The Money Flow Index (MFI) was developed by Gene Quong and Avrum Soudack. It uses both price and volume to measure buying and selling pressure.

### Parameters

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices over bar
- `low_col_name` (str) – input array of low asset price over bar
- `volume_col_name` (str) – input array of total volume over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### MidPoint indicator

```
aws.finspace.timeseries.spark.analytics.midpoint_indicator(tenor, time_col_name,
 price_col_name)
```

The Midpoint calculation is similar to the Midprice, except the highest and lowest values are returned from the same input field. The default indicator calculates the highest close and lowest close within the look back period and averages the two values.

### Parameters

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column

- `price_col_name` (str) – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**MidPrice indicator**

```
aws.finspace.timeseries.spark.analytics.midprice_indicator(tenor, time_col_name,
 high_col_name, low_col_name)
```

The Midprice returns the midpoint value from two different input fields. The default indicator calculates the highest high and lowest low within the look back period and averages the two values to return the Midprice.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Min**

```
aws.finspace.timeseries.spark.analytics.*min_indicator*(tenor, time_col_name,
 price_col_name)
```

Compute minimum value over look back period.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column

- `price_col_name` (str) – input array of prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Min and max over period**

```
aws.finspace.timeseries.spark.analytics.minmax_indicator(tenor, time_col_name,
 price_col_name)
```

Min and Max over a tenor period.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Momentum**

```
aws.finspace.timeseries.spark.analytics.momentum_indicator(tenor, time_col_name,
 input_array_col_name)
```

Compute momentum

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Moving average

```
aws.finspace.timeseries.spark.analytics.moving_average(tenor, time_col_name,
 input_array_col_name)
```

Compute moving average on window of tenor size utilizing function average_at_point

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Moving average convergence divergence (MACD)

```
aws.finspace.timeseries.spark.analytics.moving_average_converge_diverge(short_tenor,
 long_tenor, time_col_name, input_array_col_name)
```

The Moving Average Convergence Divergence (MACD) was developed by Gerald Appel, and is based on the differences between two moving averages of different lengths, a Fast and a Slow moving average. A second line, called the Signa line is plotted as a moving average of the MACD. A third line, called the MACD Histogram is optionally plotted as a histogram of the difference between the MACD and the Signal Line. Learn more.

**Parameters**

- `short_tenor` (int) – short window size
- `long_tenor` (int) – long window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Moving average convergence divergence historical (MACD)

```
aws.finspace.timeseries.spark.analytics.moving_average_converge_diverge_hist(short_tenor,
 long_tenor, signal_line_tenor, time_col_name, input_array_col_name)
```

**Parameters**

- `short_tenor` (int) – short window size
- `long_tenor` (int) – long window size
- `signal_line_tenor` (int) – signal line tenor size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Normalized average true range (NATR)

```
aws.finspace.timeseries.spark.analytics.natr_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

Normalized Average True Range (NATR) attempts to normalize the average true range values across instruments by using the formula below.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Directional movement (DM)**

```
aws.finspace.timeseries.spark.analytics.neg_dm_indicator(time_col_name, high_col_name,
 low_col_name)
```

Directional Movement (DM) is defined as the largest part of the current period price range that lies outside the previous period price range.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Negative volume index (NVI)**

```
aws.finspace.timeseries.spark.analytics.negative_volume_indicator(period,
 time_col_name, price_col_name, volume_col_name)
```

The Negative Volume Index (NVI) is a cumulative indicator that uses the change in volume to decide when the smart money is active. Paul Dysart first developed this indicator in the 1930s. Dysart's Negative Volume Index works under the assumption that the smart money is active on days when volume decreases and the not-so-smart money is active on days when volume increases.

**Parameters**

- `period`(int) – returns period typically 1
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar
- `volume_col_name` (str) – input array of total volume over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

#### Balance volume (OBV)

```
aws.finspace.timeseries.spark.analytics.on_balance_volume(time_col_name,
 price_col_name, volume_col_name)
```

On Balance Volume (OBV) maintains a cumulative running total of the amount of volume occurring on up periods compared to down periods. Learn more.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – array of asset prices
- `volume_col_name` (str) – array of volume associated with prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

#### Pairwise realized correlation

```
aws.finspace.timeseries.spark.analytics.pairwise_realized_correlation(tenor,
 time_col_name, asset1_col_name,  asset2_col_name)
```

Takes two arrays of asset prices and then produces a pairwise correlation, returns nan if the array is less than window.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `asset1_col_name` (str) – name of input array column
- `asset2_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Percent price oscillator (PPO)**

```
aws.finspace.timeseries.spark.analytics.percentage_price_oscillator(short_period,
 long_period,  signal_line_period, time_col_name, price_col_name)
```

Compute Percentage price oscillator. The Percent Price Oscillator (PPO) is based on the differences between two moving averages of different lengths, a Fast and a Slow moving average. The PPO is the difference of the two averages divided by the slower of the two moving averages, which tends to normalize the values across different instruments.

**Parameters**

- `short_period` (int) – slow ema period
- `long_period` (int) – slow ema period
- `signal_line_period` (int) – signal line ema period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Directional movement (DM)**

```
aws.finspace.timeseries.spark.analytics.pos_dm_indicator(time_col_name, high_col_name,
 low_col_name)
```

Directional Movement (DM) is defined as the largest part of the current period's price range that lies outside the previous period's price range.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Price channel**

```
aws.finspace.timeseries.spark.analytics.price_channel_indicator(tenor, time_col_name,
 price_col_name)
```

The Price Channel displays two lines, with the upper line representing the highest price and the lower line representing the lowest price for a given look back interval. The bands can optionally be smoothed with a moving average, or shifted to the right with an offset value. Basic uses of the Price Channel are to identify breakouts form the channel and to determine placement of trailing stops.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Price volume trend (PVT)**

```
aws.finspace.timeseries.spark.analytics.pvt_indicator(time_col_name, price_col_name,
 volume_col_name)
```

Compute price volume indicator. The Price Volume Trend (PVT) study attempts to quantify the amount of volume flowing into or out of an instrument by identifying the close of the period in relation to the previous period's close. The volume for the period is then allocated accordingly to a running continuous total.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `volume_col_name` (str) – input array of volumes at the asset prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Realized correlation matrix**

```
aws.finspace.timeseries.spark.analytics.realized_correlation_matrix(tenor,
 time_col_name, *kwargs)
```

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Realized volatility**

```
aws.finspace.timeseries.spark.analytics.realized_volatility(tenor, time_col_name,
 asset_price_col_name)
```

Takes in an array of asset prices and computes realized volatility. Learn more.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- **asset_price_col_name** – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Realized volatiliy spread**

```
aws.finspace.timeseries.spark.analytics.realized_volatility_spread(tenor,
 time_col_name, asset1_col_name, asset2_col_name)
```

Compute realized volatility spread between two assets.

**Parameters**

- `tenor` (int) – window size

- `time_col_name` (str) – name of time column
- `asset1_col_name` (str) – name of input array column
- `asset2_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Rate of change (ROC)**

```
aws.finspace.timeseries.spark.analytics.roc_indicator(tenor, time_col_name,
 input_array_col_name)
```

The Rate of Change (ROC) indicator compares the current price with the previous price from a selected number of periods ago. The current price is divided by the previous price and expressed as a percentage. This indicator is also commonly known as a momentum indicator. Learn more.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Relative strength index (RSI)**

```
aws.finspace.timeseries.spark.analytics.rsi(tenor, time_col_name, input_array_col_name)
```

The Relative StrengthIndex (RSI) was published by J. Welles Wilder. The current price is normalized as a percentage between 0 and 100. The name of this oscillator is misleading because it does not compare the instrument relative to another instrument or set of instruments, but rather represents the current price relative to other recent pieces within the selected look back window length. Learn more.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `input_array_col_name` (str) – name of input array column

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Parabolic stop and reverse (SAR)**

```
aws.finspace.timeseries.spark.analytics.sar_indicator(time_col_name, price_col_name,
 high_col_name, low_col_name, period=2, af=0.02, sar_rising=True)
```

The Parabolic Stop and Reverse (SAR) calculates trailing stop points to use with long and short positions. The SAR was published by J. Welles Wilderas part of a complete trend following system. The dotted lines above the price designate trailing stops for short positions; those below the price are sell stops for long positions.

**Parameters**

- `time_col_name` – name of time column
- `price_col_name` – input array of closing prices over bar
- `high_col_name` – input array of high prices over bar
- `low_col_name` – input array of low prices over bar
- `period` – window size

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Slow stock oscillator**

```
aws.finspace.timeseries.spark.analytics.slow_stock_oscillator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

The Stochastic (Stoch) normalizes price as a percentage between 0 and 100. Normally two lines are plotted, the %K line and a moving average of the %K which is called %D. A slow stochastic can be created by initially smoothing the %K line with a moving average before it is displayed. The length

of this smoothing is set in the Slow K Period. Without the initial smoothing (i.e., setting the Slow K Period to a valueof 1 ) the %K becomes the **Raw %K** value, and is also known as a fast stochastic. Learn https://www.investopedia.com/terms/s/stochasticoscillator.asp [more].

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Standard deviation indicator**

```
aws.finspace.timeseries.spark.analytics.stddev_indicator(tenor, time_col_name,
 price_col_name)
```

Compute standard deviation over a window.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Stochastic RSI (StochRSI)**

```
aws.finspace.timeseries.spark.analytics.stoch_rsi_indicator(rsi_tenor, tenor,
 time_col_name, price_col_name)
```

The Stochastic RSI (StochRSI) is an indicator used in technical analysis that ranges between zero and one (or zero and 100 on some charting platforms) and is created by applying the Stochastic oscillator formula to a set of relative strength index (RSI) values rather than to standard price data. Using RSI values within the Stochastic formula gives traders an idea of whether the current RSI value is overbought or oversold.

**Parameters**

- **rsi_tenor** (int) – window size for rsi
- `tenor` (int) – window size for stoch rsi
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Triple exponential moving average (T3)**

```
aws.finspace.timeseries.spark.analytics.t3_ema_indicator(tenor, time_col_name,
 price_col_name)
```

The Triple Exponential Moving Average (T3) by Tim Tillson attempts to offers a moving average with better smoothing then traditional exponential moving average.

`EMA1 = EMA(x,Period) EMA2 = EMA(EMA1,Period) GD = EMA1*(1+vFactor)) - (EMA2*vFactor) T3 = GD (GD ( GD(t, Period, vFactor), Period, vFactor), Period, vFactor)`

Where vFactor is a volume factor between 0 and 1 which determines how the moving averages responds. A value of 0 returns an EMA. A value of 1 returns DEMA. Tim Tillson advised or preferred a value of 0.7.

**Parameters**

- `tenor` – window size
- `time_col_name` – name of time column
- `price_col_name` – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Time series forecast (TSF)**

```
aws.finspace.timeseries.spark.analytics.time_series_forecast_indicator(tenor, scale,
 time_col_name, time_axis_col_name, price_col_name)
```

The Time Series Forecast (TSF) indicator displays the statistical trend of a security's price over a specified time period. The trend is based on linear regression analysis. Rather than plotting a straight linear regression trend line, the Time Series Forecast plots the last point of multiple linear regression trend lines. The difference between the TSF and the moving linear regression is that the TSF adds the slope of the linear regression to the linear regression essentially projecting the position of the linear regression forward one period.

**Parameters**

- `tenor` (str) – look back
- `scale` (str) – time scale, either minutes or seconds or days
- `time_col_name` (str) – name of time column
- `time_axis_col_name` (str) – name of independent time column
- `price_col_name` (str) – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**True range**

```
aws.finspace.timeseries.spark.analytics.tr_indicator(time_col_name,  price_col_name,
 high_col_name, low_col_name)
```

Calculate True Range.

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar

- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Triangular simple moving average**

```
aws.finspace.timeseries.spark.analytics.trima_indicator(tenor, time_col_name,
 price_col_name)
```

Triangular Simple Moving Average.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Triple exponential moving average (TEMA)**

```
aws.finspace.timeseries.spark.analytics.triple_exponential_moving_average(tenor,time_col_name,
 price_col_name)
```

The Triple Exponential Moving Average (TEMA) by Patrick Mulloy offers a moving average with less lag than traditional exponential moving average.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Triple exponential moving average oscillator (TRIX)**

```
aws.finspace.timeseries.spark.analytics.trix_indicator(tenor, time_col_name,
 price_col_name)
```

The Triple Exponential Moving AverageOscillator (TRIX) by Jack Hutson is a momentum indicator that oscillates around zero. It displays the percentage rate of change between two triple smoothed exponential moving averages.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Timeseries linear regression**

```
aws.finspace.timeseries.spark.analytics.ts_linear_regression(tenor, scale,
 time_col_name, time_axis_col_name, value_axis_col_name)
```

Takes two arrays, the first is the time axis, and the second is the value axis and then produces slope and intercept, where time axis is the independent variable and arr2 is the dependent variable.
**Parameters**

- `tenor` (int) – look back period
- `scale`(str) – scale of time axis, can take value of seconds, or minutes
- `time_col_name` (str) – name of time column
- `time_axis_col_name` (str) – input array of datetime
- `value_axis_col_name` (str) input array of values

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Typical price

```
aws.finspace.timeseries.spark.analytics.typical_price_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

Typical Price calculation defined as `(High + Low + Close) / 3`.

**Parameters**

- `tenor` (int) – look back window
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Ult oscillator

```
aws.finspace.timeseries.spark.analytics.ult_osc_indicator(time_col_name,
 price_col_name, high_col_name, low_col_name)
```

**Parameters**

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Weighted close price

```
aws.finspace.timeseries.spark.analytics.weighted_close_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

Weighted Close Price calculation defined as `(High+ Low + 2*Close) / 4`

**Parameters**

- `tenor` (int) – look back window
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar
- `high_col_name` (str) – input array of high prices over bar
- `low_col_name` (str) – input array of low prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Weighted linear regression

```
aws.finspace.timeseries.spark.analytics.weighted_linear_regression(tenor,
 time_col_name, input_arr1_col_name_, input_arr2_col_name_, weights_col_name)
```

Takes three arrays, the first is the independent axis, and the second is the value axis and third is weights and then produces slope and intercept.

**Parameters**

- `tenor`(str) – look back period
- `time_col_name` (str) – name of time column
- `input_arr1_col_name` – input array of independent variables
- `input_arr2_col_name` – input array of dependent variables
- `weights_col_name` (str) – input array of weights

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Weighted TS linear regression

```
aws.finspace.timeseries.spark.analytics.weighted_ts_linear_regression(tenor, scale,
 time_col_name, time_axis_col_name, value_axis_col_name, weights_axis_col_name)
```

Takes three arrays, the first is the time axis, and the second is the value axis and third is weights and then produces slope and intercept, where time axis is the independent variable and value_axis is the dependent variable, and weights.

**Parameters**

- `tenor`(str) – look back period
- `scale` (str) – scale of time axis, can take value of seconds, or minutes
- `time_col_name` (str) – name of time column
- `time_axis_col_name` (str) – input array of datetime
- `value_axis_col_name` (str) – input array of values
- `weights_axis_col_name` (str) – input array of weights

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

### Welles wilder smoothing average (WWS)

```
aws.finspace.timeseries.spark.analytics.wilder_smoothing_indicator(tenor,
 time_col_name, price_col_name)
```

The Welles Wilder's Smoothing Average (WWS) was developed by J. Welles Wilder, Jr. and is part of the Wilder's RSI indicator implementation. This indicator smoothes price movements to help you identify and spot bullish and bearish trends.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of closing prices over bar

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**%R**

```
aws.finspace.timeseries.spark.analytics.will_r_indicator(tenor, time_col_name,
 price_col_name, high_col_name, low_col_name)
```

Compute william %R indicator. The %R indicator was developed by Larry Williams and introduced in his 1979 book **How I Made $1,000,000 Trading Commodities Last Year**. Williams %R is similar to a stochastic oscillator, as it normalizes price as a percentage between 0 and 100. It is basically an inverted version of the **Raw %K** value of a Fast Stochastic.

**Parameters**

- `tenor` (int) – look back period
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of asset prices, determined by user, default is close for intraday calculations
- `high_col_name` (str) – input array of high asset prices
- `low_col_name` (str) – input array of high asset price

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Rate of change (ROC100)**

```
aws.finspace.timeseries.spark.analytics.ROC100_indicator(tenor, time_col_name,
 price_col_name)
```

The Rate of Change (ROC100) indicator compares the current price with the previous price from a selected number of periods ago. The current price is divided by the previous price and multiplied by 100. This indicator is also commonly known as a momentum indicator.

**Parameters**

- `tenor` (int) – window size

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/Spark user defined scalar function

**Rate of change percentage (ROCP)**

```
aws.finspace.timeseries.spark.analytics.ROCP_indicator(tenor, time_col_name,
 price_col_name)
```

The Rate of Change Percentage (ROCP) indicator compares the current price with the previous price from a selected number of periods ago. The current price is divided by the previous price. ROCP is not expressed as a percentage. This indicator is also commonly known as a momentum indicator.

**Parameters**

- `tenor` (int) – window size
- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/spark user defined scalar function

**Rate of change rate (ROCR)**

```
aws.finspace.timeseries.spark.analytics.ROCR_indicator(tenor, time_col_name,
 price_col_name)
```

The Rate of Change Rate (ROCR) indicator compares the current price with the previous price from a selected number of periods ago. The current price is divided by the previous price. This indicator is also commonly known as a momentum indicator.

**Parameters**

- `tenor` (int) – window size

- `time_col_name` (str) – name of time column
- `price_col_name` (str) – input array of prices

**Return type** `Callable[. . . , Column]`

**Returns** pandas/spark user defined scalar function

## Using the Amazon FinSpace library

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

This following section provides a step-by-step example on how to use the time series library across all stage in the framework, using the **US Equity TAQ 6 months, AMZN Symbol** dataset available with the sample capital markets data bundle with Amazon FinSpace.

Using the given dataset and view ids, get the view as a Spark DataFrame

```
tDF = finspace.read_view_as_spark(dataset_id = dataset_id, view_id = view_id)\
    .filter( F.col('timestamp').between('09:30:00.000000000', '16:00:00.000000000'))\
    .sort( F.col('datetime'), F.col('timestamp') )

tDF.printSchema()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…
root
 |-- timestamp: string (nullable = true)
 |-- eventtype: string (nullable = true)
 |-- ticker: string (nullable = true)
 |-- price: double (nullable = true)
 |-- quantity: long (nullable = true)
 |-- exchange: string (nullable = true)
 |-- conditions: string (nullable = true)
 |-- datetime: timestamp (nullable = true)
 |-- date: date (nullable = true)
```

**Events** – The Data View now loaded into the DataFrame contains raw data events. The DataFrame is filtered on `ticker`, `eventtype`, `datetime`, `price`, `quantity`, `exchange`, `conditions` fields.

## Interact with Spark DataFrame

As a Spark DataFrame, you can interact with the data with spark functions.

```
tDF.filter(tDF.eventtype == 'TRADE')\
    .select( 'ticker', 'eventtype', 'datetime', 'price', 'quantity', 'exchange', 'conditions')\
    .show(5, False)
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…
+------+---------+--------------------------+-------+--------+--------+----------+
|ticker|eventtype|datetime                  |price  |quantity|exchange|conditions|
+------+---------+--------------------------+-------+--------+--------+----------+
|AMZN  |TRADE    |2019-10-01 09:30:00.001207|1747.0 |15      |NASDAQ  |80000401  |
|AMZN  |TRADE    |2019-10-01 09:30:00.001269|1747.0 |21      |NASDAQ  |80000401  |
|AMZN  |TRADE    |2019-10-01 09:30:00.011732|1745.01|12      |NASDAQ  |80000401  |
|AMZN  |TRADE    |2019-10-01 09:30:00.341894|1745.74|4       |NASDAQ  |80000401  |
|AMZN  |TRADE    |2019-10-01 09:30:00.341943|1745.74|8       |NASDAQ  |a0000421  |
+------+---------+--------------------------+-------+--------+--------+----------+
only showing top 5 rows
```

**Collect Bars** – In this stage, the FinSpace create_time_bars function is used to collect raw data events into 1-minute time bars.

## Stage: Collect Bars

Collect raw TAQ events into time bars using FinSpace time series functions.

```
# define the time-bar, column for time and how much time to collect
timebar_spec   = TimeBarSpec(timestamp_column='datetime', window_duration=barWidth, slide_duration=barWidth)

# what columns to collect in the bar
bar_input_spec = BarInputSpec('activity', 'datetime', 'timestamp', 'price', 'quantity', 'exchange', 'conditions' )

# timebar column name
timebar_col = 'window'

# The results in a new DataFrame, also add column for number of activity items collected in the bar
collDF = create_time_bars(data = tDF,
                          timebar_column = timebar_col,
                          grouping_col_list = ['date', 'ticker', 'eventtype'],
                          input_spec = bar_input_spec,
                          timebar_spec = timebar_spec)\
    .withColumn( 'activity_count', F.size( F.col('activity') ) )
```

The window represents the 1-min time interval for the bar. The Activity count shows the number of events collected in each bar. Note that the data events collected inside the bar are not shown.

```
+------+---------+------------------------------------------+--------------+
|ticker|eventtype|window                                    |activity_count|
+------+---------+------------------------------------------+--------------+
|AMZN  |TRADE    |[2019-10-01 09:30:00, 2019-10-01 09:31:00]|621           |
|AMZN  |TRADE    |[2019-10-01 09:31:00, 2019-10-01 09:32:00]|451           |
|AMZN  |TRADE    |[2019-10-01 09:32:00, 2019-10-01 09:33:00]|517           |
|AMZN  |TRADE    |[2019-10-01 09:33:00, 2019-10-01 09:34:00]|1373          |
|AMZN  |TRADE    |[2019-10-01 09:34:00, 2019-10-01 09:35:00]|911           |
+------+---------+------------------------------------------+--------------+
only showing top 5 rows
```

**Summarize Bars** – In this stage, the FinSpace summarize functions are applied to calculate 1-minute summaries of events collected in bars. Summaries are created for two-point standard deviation, Volume Weighted Average Price, open(first), high, low, close(last) prices commonly referred as OHLC.

## Stage: Summarize Bars

Summarize the bars and once summarized drop activity since it will no longer be needed.

```
# Bar data is in a column that is a list of structs named 'activity'
# values collected in 'activity': datetime, teimstamp, price, quantity, exchange, conditions

sumDF = ( collDF
    .withColumn( 'std',    std( 'activity.price' ) )
    .withColumn( 'vwap',   vwap( 'activity.price', 'activity.quantity' ) )
    .withColumn( 'ohlc',   ohlc_func( 'activity.datetime', 'activity.price' ) )
    .withColumn( 'volume', total_volume( 'activity.quantity' ) )
#       .withColumn('MY_RESULT', MY_SPECIAL_FUNCTION( 'activity.datetime', 'activity.price', 'activity.quantity' ) )
    .drop( collDF.activity )
)
```

The activity count shows the number of events summarized in a single summary bar.

```
+------+---------+------------------------------------------+--------------+-----------------------------------------+------+-----------------+-----------------+
|ticker|eventtype|window                                    |activity_count|ohlc                                     |volume|vwap             |std              |
+------+---------+------------------------------------------+--------------+-----------------------------------------+------+-----------------+-----------------+
|AMZN  |TRADE    |[2019-10-01 09:30:00, 2019-10-01 09:31:00]|621           |[1747.0, 1747.0, 1743.5, 1743.61]        |38667.0|1745.8444292057827|0.6114527323064575|
|AMZN  |TRADE    |[2019-10-01 09:31:00, 2019-10-01 09:32:00]|451           |[1743.83, 1746.57, 1743.673, 1745.1057]  |7183.0|1744.9899513016844|0.5639165036598777|
|AMZN  |TRADE    |[2019-10-01 09:32:00, 2019-10-01 09:33:00]|517           |[1745.44, 1748.88, 1744.0001, 1747.9]    |8635.0|1746.92140895194  |1.3427597606260069|
|AMZN  |TRADE    |[2019-10-01 09:33:00, 2019-10-01 09:34:00]|1373          |[1747.91, 1753.803, 1747.4601, 1753.47]  |18194.0|1750.7350744696053|1.5467356344754761|
|AMZN  |TRADE    |[2019-10-01 09:34:00, 2019-10-01 09:35:00]|911           |[1753.18, 1755.5, 1746.0, 1754.29]       |13656.0|1754.2700909124196|0.717655808144455 |
+------+---------+------------------------------------------+--------------+-----------------------------------------+------+-----------------+-----------------+
only showing top 5 rows
```

**Fill & Filter** – The resulting data set is filtered according to an exchange trading calendar.

## Stage: Fill and Filter

Using the NYSE Calendar, filter values to NYSE trading days and hours.

```
# fill and filter, use the timebar defined in collect stage
ffDF = time_bar_fill_and_filter(sumDF, timebar_col, NYSECalendar20192020(), timebar_spec, start_date, end_date)

ffDF = sumDF

# note, times are UTC after running file and filter
ffDF.filter( (ffDF.date == start_date) & (ffDF.eventtype == 'TRADE') )\
    .filter( ffDF.ticker == 'AMZN' )\
    .sort( ffDF.window )\
    .select( 'ticker', 'eventtype', 'window', 'activity_count', 'ohlc', 'volume', 'vwap', 'std' )\
    .show(5, False)
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…
+------+---------+-----------------------------------------+--------------+-----------------------------------+-------+-----------------+-----------------+
|ticker|eventtype|window                                   |activity_count|ohlc                               |volume |vwap             |std              |
+------+---------+-----------------------------------------+--------------+-----------------------------------+-------+-----------------+-----------------+
|AMZN  |TRADE    |[2019-10-01 09:30:00, 2019-10-01 09:31:00]|621           |[1747.0, 1747.0, 1743.5, 1743.61]  |38667.0|1745.8444292057827|0.6114527323064575|
|AMZN  |TRADE    |[2019-10-01 09:31:00, 2019-10-01 09:32:00]|451           |[1743.83, 1746.57, 1743.673, 1745.1057]|7183.0 |1744.9899513016844|0.5639165036598777|
|AMZN  |TRADE    |[2019-10-01 09:32:00, 2019-10-01 09:33:00]|517           |[1745.44, 1748.88, 1744.0001, 1747.9]|8635.0 |1746.92140895194 |1.3427597606260069|
|AMZN  |TRADE    |[2019-10-01 09:33:00, 2019-10-01 09:34:00]|1373          |[1747.91, 1753.803, 1747.4601, 1753.47]|18194.0|1750.7350744696053|1.5467356344754761|
|AMZN  |TRADE    |[2019-10-01 09:34:00, 2019-10-01 09:35:00]|911           |[1753.18, 1755.5, 1746.0, 1754.29] |13656.0|1754.2700909124196|0.717655808144455 |
+------+---------+-----------------------------------------+--------------+-----------------------------------+-------+-----------------+-----------------+
only showing top 5 rows
```

The schema is simplified to prepare a dataset of features. VWAP and standard deviation calculations are now displayed as well.

```
# sample the data
prepDF.filter( (prepDF.date == start_date) & (prepDF.eventtype == 'TRADE') )\
    .filter( prepDF.ticker == 'AMZN' )\
    .sort( prepDF.start )\
    .select( 'ticker', 'eventtype', 'start', 'end', 'activity_count', 'std', 'vwap', 'open', 'high', 'low', 'close' , 'volume' )\
    .show(10, False)
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…
+------+---------+-------------------+-------------------+--------------+-------------------+-----------------+-------+-------+--------+--------+-------+
|ticker|eventtype|start              |end                |activity_count|std                |vwap             |open   |high   |low     |close   |volume |
+------+---------+-------------------+-------------------+--------------+-------------------+-----------------+-------+-------+--------+--------+-------+
|AMZN  |TRADE    |2019-10-01 09:30:00|2019-10-01 09:31:00|621           |0.6114527323064575 |1745.8444292057827|1747.0 |1747.0 |1743.5  |1743.61 |38667.0|
|AMZN  |TRADE    |2019-10-01 09:31:00|2019-10-01 09:32:00|451           |0.5639165036598777 |1744.9899513016844|1743.83|1746.57|1743.673|1745.1057|7183.0 |
|AMZN  |TRADE    |2019-10-01 09:32:00|2019-10-01 09:33:00|517           |1.3427597606260069 |1746.92140895194 |1745.44|1748.88|1744.0001|1747.9  |8635.0 |
|AMZN  |TRADE    |2019-10-01 09:33:00|2019-10-01 09:34:00|1373          |1.5467356344754761 |1750.7350744696053|1747.91|1753.803|1747.4601|1753.47 |18194.0|
|AMZN  |TRADE    |2019-10-01 09:34:00|2019-10-01 09:35:00|911           |0.717655808144455  |1754.2700909124196|1753.18|1755.5 |1746.0  |1754.29 |13656.0|
|AMZN  |TRADE    |2019-10-01 09:35:00|2019-10-01 09:36:00|547           |0.3888813336732253 |1754.1663939495982|1754.29|1755.1 |1753.212|1753.62 |9206.0 |
|AMZN  |TRADE    |2019-10-01 09:36:00|2019-10-01 09:37:00|595           |0.3609497389792301 |1754.4009736560563|1753.62|1755.55|1753.06 |1754.8381|18881.0|
|AMZN  |TRADE    |2019-10-01 09:37:00|2019-10-01 09:38:00|413           |0.39938420294566096|1754.909071567436 |1754.84|1755.66|1754.0  |1754.08 |6584.0 |
|AMZN  |TRADE    |2019-10-01 09:38:00|2019-10-01 09:39:00|371           |0.4020568446856403 |1753.1713176197643|1754.08|1754.57|1752.52 |1753.63 |7974.0 |
|AMZN  |TRADE    |2019-10-01 09:39:00|2019-10-01 09:40:00|418           |0.36652483968841376|1753.6643422187742|1753.51|1754.48|1752.59 |1753.425|6445.0 |
+------+---------+-------------------+-------------------+--------------+-------------------+-----------------+-------+-------+--------+--------+-------+
only showing top 10 rows
```

**Apply Analytics** – FinSpace Bollinger Bands function is applied on the features dataset. Note that the tenor window to perform the calculation is 15 which means that the calculation is applied when 15 data events are available. As each event corresponds to a 1-min summary bar in the features dataset, the resulting dataset starts from timestamp 09:45 (see end column).

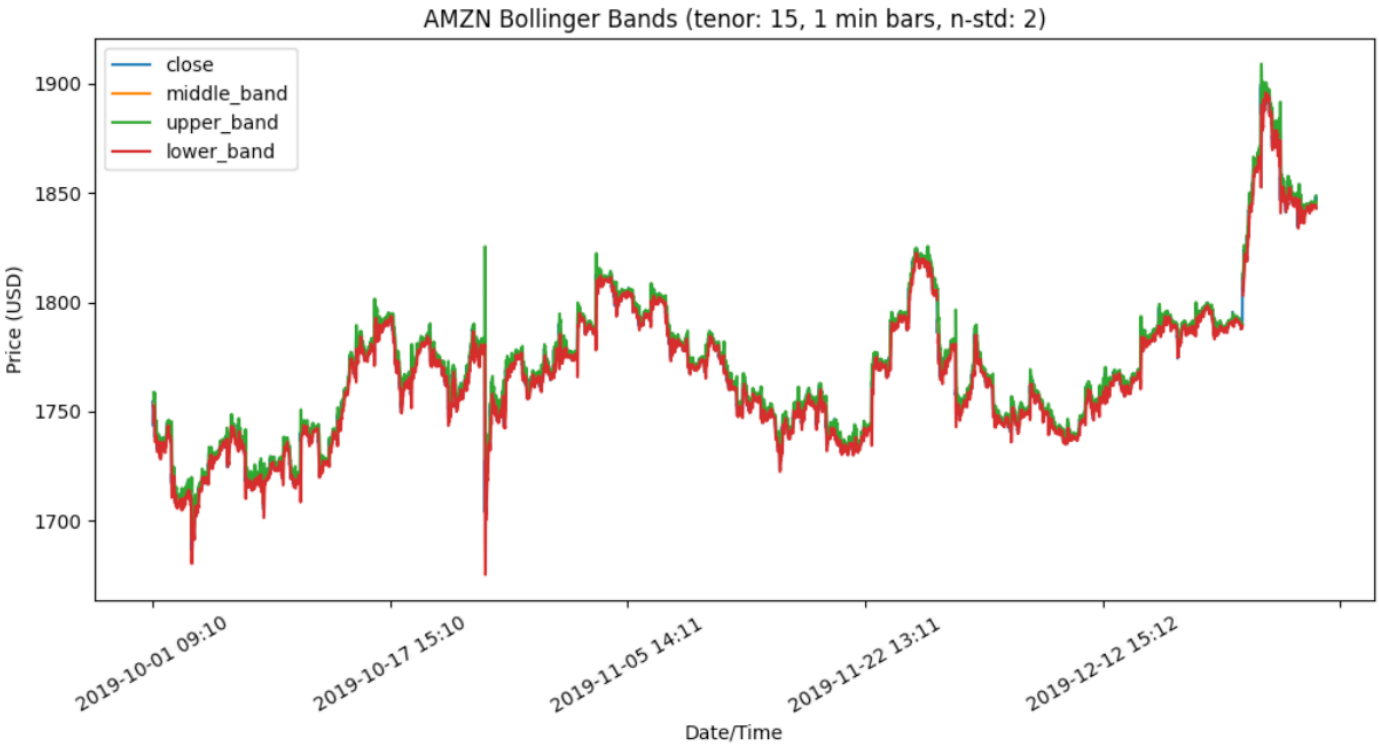
## Stage: Apply Analytics

Now apply analytics to the data, in our case calculate realized volatility and bollinger bands

```
# Arguments to the functions
tenor = 15
numStd = 2

# analytics to calculate
bbandsDef  = bollinger_bands(tenor, numStd, "end", "vwap", "high", "low")

# group the dataset's values by....
partitionList = ["ticker", "eventtype"]

# Prepare the dataframe
tsDF = prepDF

tsDF = compute_analytics_on_features(tsDF, "bollinger_band", bbandsDef, partition_col_list = partitionList)

# will be working with the once calculated, lets cache it
tsDF = tsDF.cache()

tsDF.printSchema()
```

```
+------+---------+-----------------+-----------------+---------+---------+-----------------------------------------------------------+
|ticker|eventtype|start            |end              |open     |close    |bollinger_band                                             |
+------+---------+-----------------+-----------------+---------+---------+-----------------------------------------------------------+
|AMZN  |TRADE    |2019-10-01 09:44:00|2019-10-01 09:45:00|1754.29  |1753.88  |[1758.1393614009828, 1751.7329893524907, 1745.3266173039985]|
|AMZN  |TRADE    |2019-10-01 09:45:00|2019-10-01 09:46:00|1754.32  |1753.8   |[1757.9034002206815, 1752.3328159840662, 1746.7622317474509]|
|AMZN  |TRADE    |2019-10-01 09:46:00|2019-10-01 09:47:00|1753.99  |1752.93  |[1756.8821058854173, 1752.8780021131315, 1748.8738983408457]|
|AMZN  |TRADE    |2019-10-01 09:47:00|2019-10-01 09:48:00|1752.94  |1754.537 |[1755.5477463075304, 1753.3534823449488, 1751.1592183823673]|
|AMZN  |TRADE    |2019-10-01 09:48:00|2019-10-01 09:49:00|1754.4   |1755.0506|[1755.1915870635785, 1753.5036250999726, 1751.8156631363668]|
|AMZN  |TRADE    |2019-10-01 09:49:00|2019-10-01 09:50:00|1755.39  |1754.0   |[1755.1972130747945, 1753.5015939911837, 1751.8059749075728]|
|AMZN  |TRADE    |2019-10-01 09:50:00|2019-10-01 09:51:00|1754.3468|1752.0499|[1755.0984835030401, 1753.4250749429034, 1751.7516663827666]|
|AMZN  |TRADE    |2019-10-01 09:51:00|2019-10-01 09:52:00|1751.87  |1751.4904|[1755.0935206292484, 1753.233334517757, 1751.3731484062655] |
|AMZN  |TRADE    |2019-10-01 09:52:00|2019-10-01 09:53:00|1751.4202|1750.8   |[1754.967472049993, 1752.9648841045178, 1750.9622961590426] |
|AMZN  |TRADE    |2019-10-01 09:53:00|2019-10-01 09:54:00|1750.57  |1746.0   |[1755.5445614104572, 1752.6514291752815, 1749.7582969401058]|
+------+---------+-----------------+-----------------+---------+---------+-----------------------------------------------------------+
only showing top 10 rows
```

You can plot the output into a chart using matplotlib. The chart shows the Bollinger Bands for the entire 3 month history for AMZN.




FinSpace time series library is provided with `aws.finspace.timeseries.spark` package used when working with data that will be processed using a FinSpace Spark cluster in the FinSpace notebook.

# Amazon FinSpace administration

Use the following section to learn about Amazon FinSpace administrative tasks.

**Topics**



## Regions and IP ranges

AWS cloud-computing resources are housed in highly available facilities in different areas of the world (for example, North America, Europe, and Asia). These facilities are each part of an AWS Region. For more information about Regions and AZs, see Global infrastructure.

The following table provides supported regions and endpoints for Amazon FinSpace Managed kdb Insights.

| Region name | Region code | Endpoint (HTTPS) |
|---|---|---|
| US East (Ohio) | us-east-2 | finspace.us-east-2.amazonaws.com |
| US East (N. Virginia) | us-east-1 | finspace.us-east-1.amazonaws.com |
| US West (Oregon) | us-west-2 | finspace.us-west-2.amazonaws.com |
| Canada (Central) | ca-central-1 | finspace.ca-central-1.amazonaws.com |
| Europe (Ireland) | eu-west-1 | finspace.eu-west-1.amazonaws.com |
| Europe (London) | eu-west-2 | finspace.eu-west-2.amazonaws.com |

| Region name | Region code | Endpoint (HTTPS) |
|---|---|---|
| Europe (Frankfurt) | eu-central-1 | finspace.eu-central-1.amazonaws.com |
| Asia Pacific (Singapore) | ap-southeast-1 | finspace.ap-southeast-1.amazonaws.com |
| Asia Pacific (Sydney) | ap-southeast-2 | finspace.ap-southeast-2.amazonaws.com |
| Asia Pacific (Tokyo) | ap-northeast-1 | finspace.ap-northeast-1.amazonaws.com |

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The following table provides supported regions and endpoints for Amazon FinSpace Dataset Browser.

| Region name | Region code | Endpoint (HTTPS) |
|---|---|---|
| US East (Ohio) | us-east-2 | finspace.us-east-2.amazonaws.com |
| US East (N. Virginia) | us-east-1 | finspace.us-east-1.amazonaws.com |
| US West (Oregon) | us-west-2 | finspace.us-west-2.amazonaws.com |

| Region name | Region code | Endpoint (HTTPS) |
| --- | --- | --- |
| Canada (Central) | ca-central-1 | finspace.ca-central-1.amazonaws.com |
| Europe (Ireland) | eu-west-1 | finspace.eu-west-1.amazonaws.com |

# Supported browsers

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Before you start using the FinSpace web application, use the following table to verify that your browser is supported.

| Browser | Version | Check your version |
| --- | --- | --- |
| **Apple Safari for macOS** | Latest three major versions | Open Safari. On the menu, choose **Safari**, and then choose **About Safari**. The version number is shown in the dialog box that displays. |
| **Google Chrome** | Latest three major versions | Open Chrome and type `chrome://version` in your address bar. The version is in the **Google Chrome** field at the top of the results. |

| Browser | Version | Check your version |
| --- | --- | --- |
| **Microsoft Edge** | Latest three major versions | Open Microsoft Edge. Select Settings and more at the top of the window, and then select Settings. Scroll down and select About Microsoft Edge. |
| **Mozilla Firefox** | Latest three major versions | Open Firefox. On the menu, choose the Help icon, and then choose About Firefox. The version number is listed underneath the Firefox name. |

# Amazon FinSpace security

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

1. **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to FinSpace, see AWS services in scope by compliance program.
2. **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using FinSpace. The following topics show you how to configure FinSpace to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your FinSpace resources.

**Topics**

# Identity and access management in Amazon FinSpace

This section explains the identity management and authentication for Amazon FinSpace Managed kdb and Dataset browser.

## Identity management for Managed kdb

Amazon FinSpace Managed kdb uses AWS Identity and Access Management (IAM) policies to restrict access to operations.

Whenever you use IAM policies, ensure that you follow IAM best practices. For more information, see Security best practices in the *IAM User Guide*.

## Identity management for Dataset browser

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace Dataset browser supports two methods for identity management and authentication. A FinSpace dataset browser environment can be created with either of the following methods.

1. **Email and password** – FinSpace access is controlled via users that are created and managed within the FinSpace application. With email and password based authentication method, users sign in to FinSpace using their email address and password. An environment created with email and password based authentication method cannot be changed to SSO based authentication method in the future. Learn more about Managing user access with email and password.
2. **Single Sign-On (SSO)** – FinSpace access is controlled through your organization's identity provider (IdP). With this authentication method, users will be redirected to the SSO login page of their Security Assertion Markup Language 2.0 (SAML 2.0) compliant identity provider (IdP) solution to authenticate their access to FinSpace. An environment created with SSO based

authentication method cannot be changed to email and password based authentication method in the future. Learn more about creating and managing users with SAML based SSO.

**Topics**



## Setting up SAML based single sign-on (SSO) with Amazon FinSpace

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

When you use SAML based SSO, you can manage users with your enterprise identity provider (IdP). You can use a third-party identity provider that supports through Security Assertion Markup Language 2.0 (SAML 2.0) to provide a simple on-boarding flow for your Amazon FinSpace users. Such identity providers include Microsoft Windows Active Directory Federation Services and Okta among others.

With SSO, your users get one-click access to their FinSpace applications using their existing identity credentials. You also have the security benefit of identity authentication by your identity provider. You can control which users have access to FinSpace using your existing identity provider.

**Topics**

# Tutorial: Setup an Identity Provider with your Amazon FinSpace environment

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can integrate any SAML 2.0 compliant IdP when creating a new Amazon FinSpace environment.

**Prerequisites**

Before creating a FinSpace environment with SAML based SSO, do the following:

Inside your organization's network, configure your identity store, such as Windows Active Directory, to work with a SAML-based IdP. SAML based IdPs include Microsoft Windows Active Directory Federation Services, Okta, and so on.

**Step 1: Generate a SAML metadata document**

Using your IdP, generate a metadata document that describes your organization as an identity provider. You will need the metadata document or the URL to the metadata document when creating the FinSpace environment.

**Step 2: Determine the SAML attribute for email**

Determine the SAML attribute name that contains the email address in the SAML assertion. Email address is required to identify the user in FinSpace. For example, `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`. Check your IdP documentation for details. You will need the SAML attribute when creating the FinSpace environment.

**Step 3: Create a FinSpace environment**

Create a FinSpace environment. Once the FinSpace environment is ready, copy and save the **Redirect / Sign-in url** and **URN** from the Summary section of the environment page. You will need the parameters for configuration in the IdP.

**Step 4: Create an application for FinSpace in your IdP**

Once the environment is created, add an application for FinSpace in your IdP and use the **Redirect / Sign-in url** and **URN** where appropriate.

**Step 5: Assign users to the newly created FinSpace application in your IdP**

Once the application is added, assign users to the application in IdP. A minimum of one user is required to create a superuser in FinSpace.

**Step 6: Create a superuser in your FinSpace environment**

> **Note**
>
> In order to create a FinSpace environment, you need to be a user with **AdministratorAccess** role or FinSpace policy.

Now that the users are assigned to your FinSpace application in your IdP, create a superuser.

After your FinSpace is created, you must create a first superuser to add additional users and to configure permission groups from within the FinSpace web application. A superuser has all permissions to take all actions in FinSpace. The first superuser must be created in the AWS console page. After the superuser is created, the superuser logs in to the FinSpace web application for the first time.

**To create a superuser**

1. Sign in to your AWS account in which the FinSpace environment was created and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace. Your AWS account number is displayed for verification purposes.
2. Choose **Environments** and select the FinSpace environment for which a superuser will be created.
3. Under **Superusers**, choose **Add Superuser**.
4. On **Specify Superuser details** page, enter the **Email address**, **First name**, and **Last name**.
5. Choose **Next**.
6. On the next page, review the superuser details.
7. Choose **Create and view credentials** to get a temporary password.

> **Note**
>
> If you have created an environment with SSO, you will not get a temporary password as you will be authenticated with your IdP.

8. On the **View Credentials** page, view and copy the superuser security credentials. You also get a welcome message which you can use to email users instructions for signing into FinSpace.

   Share these credentials with the person designated as the superuser. The credentials are necessary to sign in to your FinSpace web application. The **Environment domain** is the sign-in url for your FinSpace web application.

   > **Note**
   >
   > This is the last time these credentials will be available to be copied. However, you can create new credentials at any time.

You have successfully created a FinSpace environment configured with your SAML 2.0 IdP. Learn more about managing users in SSO and permissions.

## Tutorial: Creating an Amazon FinSpace environment with Okta SSO

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The following tutorial walks you through how Amazon FinSpace environment can be created using Okta as an Identity provider (IdP).

### Prerequisites

Ensure that a user exists in Okta for each person who will need access to FinSpace. When creating users, make sure to include an email address for each user. Email addresses are required to connect the users in Active Directory Federation Services with their corresponding users in FinSpace.

### Step 1: Creating an Okta application

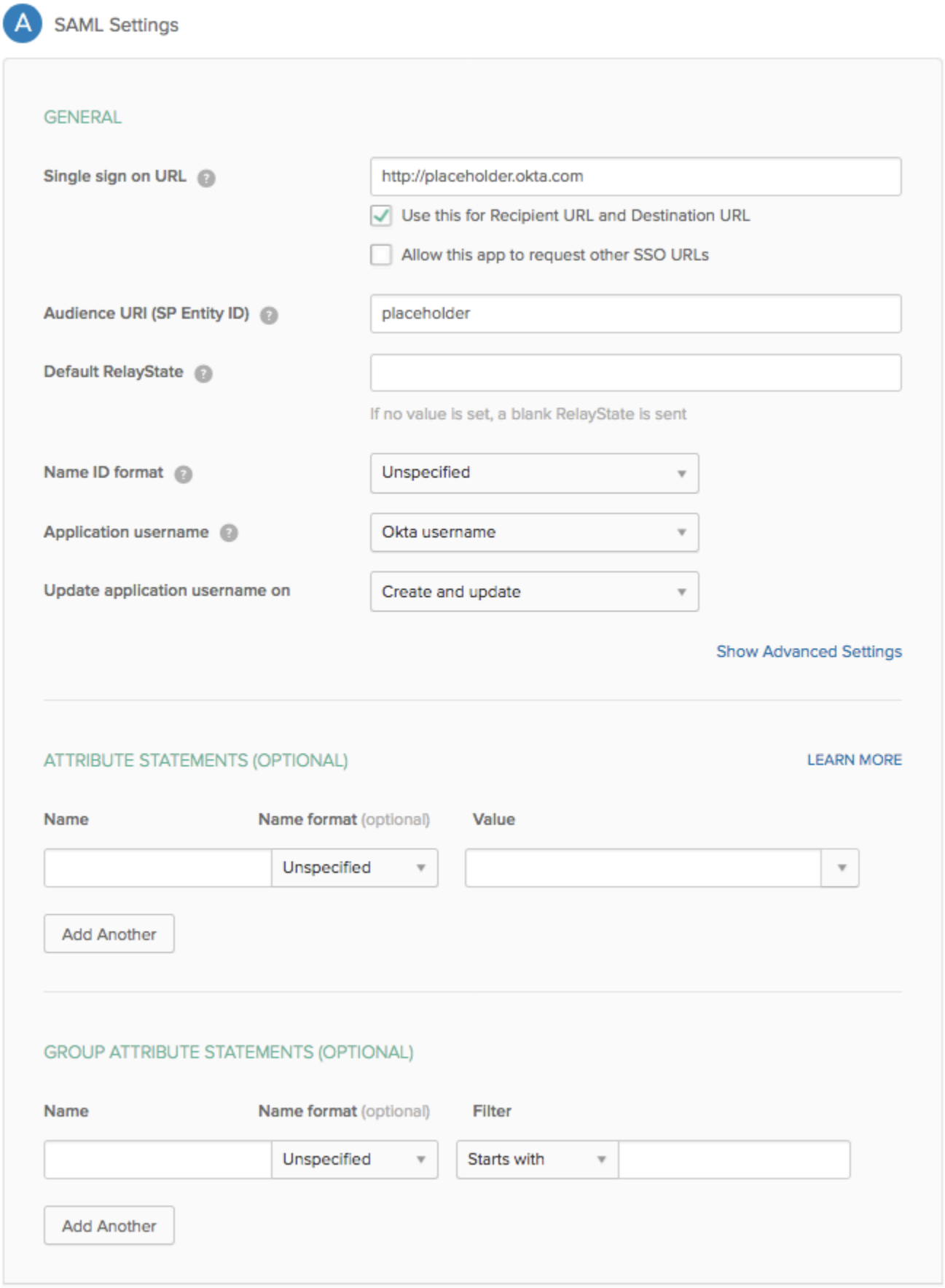
> **Note**
>
> You need to have administrator privileges in Okta for this tutorial.

### To create an Okta application

1. Sign in to your Okta admin dashboard.

   If you don't have an account, you can create a free Okta developer edition account.
2. Choose **Applications**.
3. Choose **Add Application**.
4. Choose **Create New App**.
5. On the **Create New Application Integration** page, for **Platform** select **Web** from the drop down menu.
6. For **Sign in method**, choose **SAML 2.0** and then choose **Create**.
7. Specify an **App name**. For example, `FinSpace`.
8. Choose **Next**.
9. For the **Single sign on URL**, use `http://placeholder.okta.com` .

   > **Note**
   >
   > This is just a placeholder url to generate the SAML meta data document. You will get the actual single sign on URL once FinSpace environment is created.


A
SAML Settings
GENERAL
Single sign on URL
http://placeholder.okta.com
Use this for Recipient URL and Destination URL
Allow this app to request other SSO URLs
Audience URI (SP Entity ID)
placeholder
Default RelayState
If no value is set, a blank RelayState is sent
Name ID format
Unspecified
Application username
Okta username
Update application username on
Create and update
Show Advanced Settings
ATTRIBUTE STATEMENTS (OPTIONAL)
LEARN MORE
Name
Name format (optional)
Value
Unspecified
Add Another
GROUP ATTRIBUTE STATEMENTS (OPTIONAL)
Name
Name format (optional)
Filter
Unspecified
Starts with
Add Another

10. For **Audience URI (SP Entity ID)**, enter `placeholder`.

> **Note**
>
> This is just a placeholder Uniform Resource Name (URN) to generate the SAML meta data doc. You will get the actual URN once FinSpace environment is created.

11. Under **ATTRIBUTE STATEMENTS** section, enter the following:
    a. **Name** – `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`
    b. **Value** – `user.email`
12. Choose **Next**.
13. Choose **I'm an Okta customer adding an internal app**.
14. Choose **Finish**.
15. Choose **Identity Provider metadata** and then choose **Copy Link Address**.
16. Save the link to a notepad. You can also choose to save SAML metadata document instead of the link.

Now that you have the SAML metadata document or its URL, let's create a FinSpace environment.

**Step 2: Creating a FinSpace environment**

**To create a FinSpace environment**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Create Environment**.
3. Enter a name for your FinSpace environment under **Environment name**. For example, enter `finspace-saml-okta`
4. (Optional) Add **Environment description**.
5. Select an existing or create a new KMS key to encrypt data in your FinSpace environment. For more information, see Managing keys.
6. For **Authentication method**, select **Single Sign On (SSO)**.
7. Enter your **Identity provider name**. For example, `Okta`.

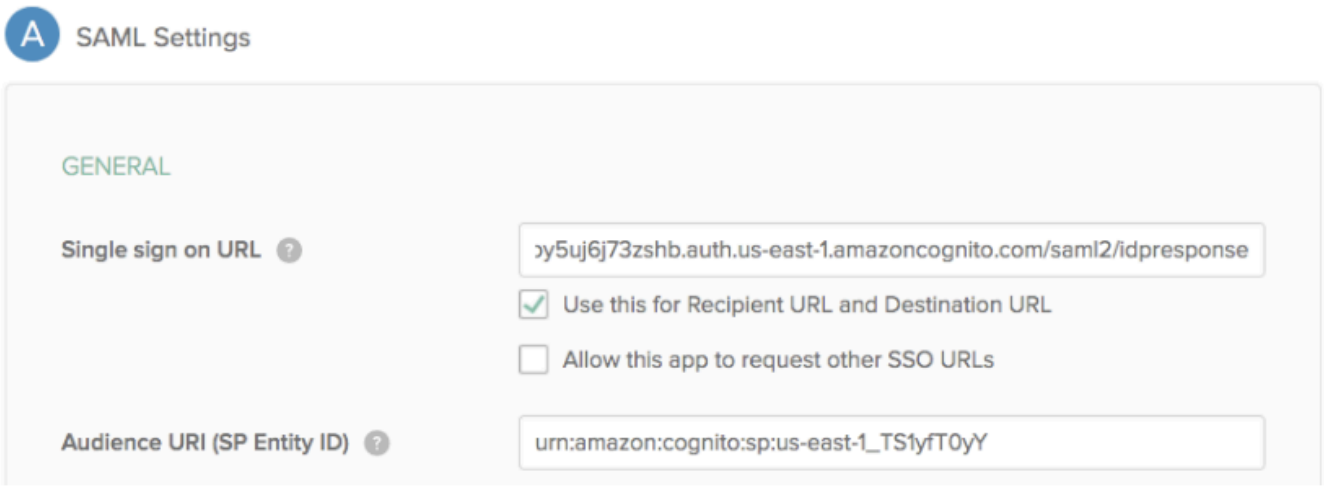
8. For **Metadata document URL**, select **Provide a metadata document URL** and then paste the SAML metadata document URL in the text box.
9. For **Attribute mapping**, enter the attribute set for email in Okta. Since you set email attribute as `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`, the same value should be set in this field.
10. Under **Initial Superuser**, enter the details to setup the first superuser.
11. Choose **Create Environment**. The environment creation process starts and it will take 50-60 minutes to finish in the background. You can return to other activities while the environment is being created.
12. After the FinSpace environment is ready, copy and save the **Redirect / Sign-in URL** and **URN**.

Your FinSpace is now created. Finish configuration in Okta.

**Step 3: Finish application configuration in Okta**

Finish configuration of your FinSpace Okta app with the **Redirect / Sign-in URL** and **URN**.

1. Sign in to your Okta console.
2. Choose **Admin** on the top-right corner.
3. From the top bar menu bar, choose **Applications**.
4. Choose the **FinSpace** app that you had setup with placeholders.
5. Under the **General** tab, scroll to **General Settings** and choose **Edit** on SAML settings.
6. Choose **Next**.
7. For **Single Sign On URL**, paste the copied **Redirect / Sign-in URL** from FinSpace environment.
8. Select the **Use this for Recipient URL and Destination URL** check box.
9. For **Audience URI (SP Entity ID)**, enter the copied **URN** from the FinSpace environment.




10. Choose **Next**.
11. Choose **Finish**.

### Step 4: Assign user to the FinSpace application in Okta

Now that the application is setup. Assign at least one user to the FinSpace app in Okta who can be created as a superuser for FinSpace.

**To assign user to the FinSpace application in Okta**

1. Sign in to your Okta console.
2. Choose **Admin** on the top-right corner.
3. From the top bar menu bar, choose **Applications**.
4. Choose the **FinSpace**.
5. Choose the **Assignments** tab.
6. Choose the **Assign** drop down menu. A list of users appears.
7. Choose **Assign next** for the user that you want to designate as the superuser in FinSpace. You may add multiple users at this point too.
8. Choose **Save and Go back**.

### Step 5: Create superuser in your FinSpace environment

Now that a user is assigned, they can be created as a superuser in FinSpace.

**To create a superuser**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose `finspace-saml-okta` from the list of environments.
3. Under **Superusers**, choose **Add Superuser**.
4. On **Specify Superuser details** page, enter the email that was used when assigning the user in Okta.
5. Enter the **First name** and the **Last name**.
6. Choose **Create and view credentials**. You will not receive a password as you will use the Okta Idp credentials for authentication.

### Step 6: Sign in to FinSpace with Okta IdP credentials

**To sign in with Okta IdP credentials**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose `finspace-saml-okta` from the list of environments.
3. Copy the link under **Environment domain** and paste it in your web browser.

   You will be re-directed to your Okta Idp authentication page.
4. Enter your SSO credentials to sign in to FinSpace.

## Tutorial: Creating an Amazon FinSpace environment with IAM Identity Center

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The following tutorial walks you through how FinSpace environment can be created using AWS IAM Identity Center as an Identity provider (IdP).

**Prerequisites**

Ensure that a user exists in IAM Identity Center for each person who will need access to FinSpace. When creating users, make sure to include an email address for each user. Email addresses are required to connect the users in Active Directory Federation Services with their corresponding users in FinSpace.

**Step 1: Creating an application in IAM Identity Center**

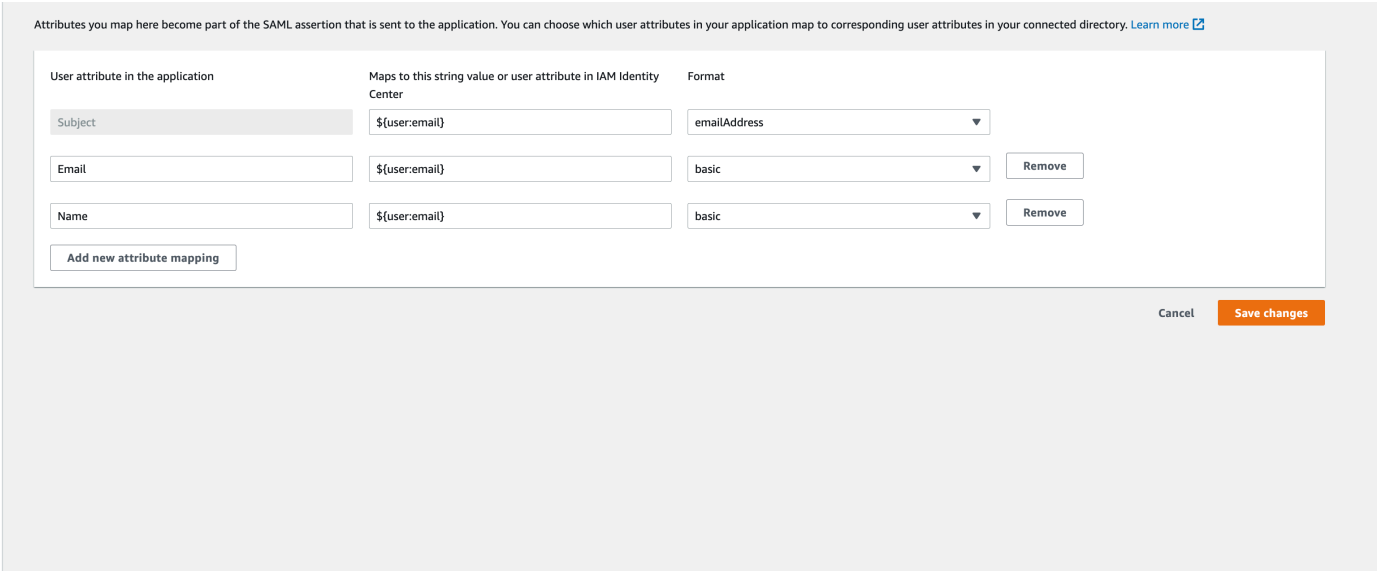
> **Note**
>
> You need to have appropriate privileges in IAM Identity Center to create a SAML application.

**To create an application in IAM Identity Center**

1. Sign in to AWS Management Console, and open IAM Identity Center.
2. Choose **Settings**.
3. For **Identity source**, choose **IAM Identity Center**.
4. From the left menu, choose **Applications**.
5. Choose **Add application**.
6. Choose **Add a custom SAML 2.0 application**.
7. Choose **Next**.
8. On the **Configure application** page, specify a display name for the application. For example, you can use `FinSpace-SAML-application`.
9. (Optional) Add a description.
10. Copy and save the URL for **IAM Identity Center SAML metadata file** or download it. You will need it when you create a FinSpace environment.
11. For **Application metadata**, choose **Manually type your metadata values**.
12. For **Application ACS URL**, enter `https://finspace.com/saml2/idpresponse`. For **Application SAML audience**, enter `urn:amazon:sp:*`.

> **Note**
>
> These are sample values. Return to application configuration and replace these fields with the actual values after you create an environment.

13. Choose **Submit**. The page for newly created application opens.
14. On the application page, choose **Actions** and then choose **Edit attribute mappings**.
15. On the attribute mappings page, enter the attribute mappings values as shown in the following screenshot.




16. Choose **Save changes**.

Now that you have the SAML metadata document or it's URL, create a FinSpace environment next.

### Step 2: Creating a FinSpace environment

### To create a FinSpace environment

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Create Environment**.
3. Enter a name for your FinSpace environment under **Environment name**. For example, enter `finspace-saml-aws-sso`
4. (Optional) Add **Environment description**.

5. Select an existing or create a new KMS key to encrypt data in your FinSpace environment. For more information, see Managing keys.
6. For **Authentication method**, select **Single Sign On (SSO)**.
7. Enter your **Identity provider name**. For example, **`IAM Identity Center`**.
8. For **Metadata document URL**, choose **Provide a metadata document URL** and then paste the SAML metadata document URL in the text box. This is the same URL that you copied when creating an application.
9. For **Attribute mapping**, enter the attribute set for email in IAM Identity Center. Since you set attribute as `Email` in SSO, set the same in mapping.
10. Choose **Create Environment**. The environment creation process starts and it will take 50-60 minutes to finish in the background. You can return to other activities while the environment is being created.
11. After the FinSpace environment is ready, copy and save the **Redirect / Sign-in URL** and **URN**.

#### Step 3: Finish application configuration in IAM Identity Center

Finish configuration of IAM Identity Center app with the **Redirect / Sign-in URL** and **URN**.

1. Sign in to AWS Management Console, and open IAM Identity Center.
2. Choose **Applications**.
3. Choose **FinSpace-SAML-application** that you created in step 1 of this tutorial.
4. On the application details page, choose **Actions** and then choose **Edit configuration**.
5. In the **Application metadata** section, paste the following values that you copied in step 2 of this tutorial.
    a. For **Application ACS URL**, paste the **Redirect / Sign-in URL**.
    b. For **Application SAML audience**, paste the **URN**.
6. Choose **Submit**.

#### Step 4: Assign user to the FinSpace application in IAM Identity Center

After setting up the application, assign at least one user to it in IAM Identity Center. You can create this user as a superuser for FinSpace.

**To assign a user**

1. Sign in to AWS Management Console, and open IAM Identity Center.
2. Choose **Applications**.
3. Choose the `FinSpace-SAML-application` application.
4. Choose **Assign Users**.
5. From the list of users, choose and assign users to the application.

### Step 5: Create superuser in your FinSpace environment

After assigning a user,you can create them as a superuser in FinSpace.

**To create a superuser**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose `finspace-saml-aws-sso` from the list of environments.
3. Under **Superusers**, choose **Add Superuser**.
4. On the **Specify Superuser details** page, enter the email that was used when assigning the user in IAM Identity Center.
5. Enter the **First name** and the **Last name**.
6. Choose **Next**.
7. Review the details and choose **Create and view credentials**. You will not receive a password as you will use the IAM Identity Center credentials for authentication.

### Step 6: Sign in to FinSpace with IAM Identity Center credentials

**To sign in with IAM Identity Center credentials**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose `finspace-saml-aws-sso` from the list of environments.
3. Choose the **Application URL** link.

   The IAM Identity Center authentication page opens.

4. Enter your SSO credentials to sign in to FinSpace.

## Tutorial: Creating an Amazon FinSpace environment with AD FS

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

The following tutorial walks you through how Amazon FinSpace environment can be created using Microsoft Active Directory Federation Services (AD FS) as an Identity provider (IdP).

> **Note**
>
> You need to have appropriate privileges in AD FS to create a SAML application.

### Prerequisites

Ensure that a user exists in AD FS for each person who will need access to FinSpace. When creating users, make sure to include an email address for each user. Email addresses are required to connect the users in AD FS with their corresponding users in FinSpace.

### Step 1: Access the SAML metadata document or URL from AD FS

Access the SAML metadata document or URL from your AD FS installation. You will need this document or URL to create the FinSpace environment.

### Step 2: Creating a FinSpace environment

**To create a FinSpace environment**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose **Create Environment**.

3. Enter a name for your FinSpace environment under **Environment name**. For example, enter `finspace-saml-adfs`.
4. (Optional) Add **Environment description**.
5. Select an existing or create a new KMS key to encrypt data in your FinSpace environment. For more information, see Managing keys.
6. For **Authentication method**, select **Single Sign On (SSO)**.
7. Enter your **Identity provider name**. For example, `AD FS`.
8. For **Metadata document URL**, select **Provide a metadata document URL** and then paste the SAML metadata document URL in the text box.
9. For **Attribute mapping**, enter the attribute set for email in AD FS. It should be `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`.
10. Choose **Create Environment**. The environment creation process starts and it will take 50-60 minutes to finish in the background. You can return to other activities while the environment is being created.
11. After the FinSpace environment is ready, copy and save the **Redirect / Sign-in URL** and **URN**.

**Step 3: Configure AD FS for FinSpace**

**To configure ADFS for FinSpace**

1. Sign in to your AD FS console.
2. Go to **Server Manager**.
3. From the top-right drop down menu, choose **Tools**.
4. Choose **AD FS management**.
5. From the left menu, choose **Relying Party Trusts**.
6. Choose **Add Relying Party Trust**.
7. From the dialog box, choose **Claims Aware**.
8. Choose **Enter data about the relying party manually**.
9. For display name, enter `FinSpace` and then choose **Next**.
10. Choose **Enable support for the SAML 2.0 WebSSO protocol**.
11. Paste the **Redirect / Sign-in URL** and then choose **Next**.
12. Paste the **URN** under the **Relying party trust identifier**.

13. Choose **Add** and then choose **Next**.
14. Choose **Close**. You will see **FinSpace** in the list of **Relying Party Trusts**.
15. Right-click on **FinSpace** and choose **Edit Claim Issuance Policy**.
16. On the next page, chose **Add Rule**.
17. Under **Claim Rule Template**, choose **Send LDAP Attributes as Claims**.
18. Choose **Next**.
19. For **Claim rule name**, enter rule name as `emailclaimrule`.
20. Under **Attribute store**, choose **Active Directory**.
21. Under **Mapping of LDAP attributes to outgoing claim types**, set the LDAP attributes as following:
    a. For **LDAP attribute**, enter `E-mail-Addresses` and for **Outgoing Claim Type** , enter `E-mail Address`.
    b. Repeat the above step to set **LDAP attribute**, as `E-mail-Addresses` and **Outgoing Claim Type** as `Name ID`.
22. Choose **Finish** and then choose **OK**.

#### Step 4: Assign user in AD FS

Ensure that any user to be enabled for FinSpace has a valid email in their user record in AD FS.

#### Step 5: Create superuser in your FinSpace environment

#### To create a superuser

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose `finspace-saml-adfs` from the list of environments.
3. Under **Superusers**, choose **Add Superuser**.
4. On **Specify Superuser details** page, enter the email that was used when assigning the user in AD FS.
5. Enter the **First name** and the **Last name**.
6. Choose **Create and view credentials**. You will not receive a password as you will use the IAM Identity Center credentials for authentication.

**Step 6: Sign in to FinSpace with AWS SSO credentials**

**To sign in with IAM Identity Center credentials**

1. Sign in to the AWS Management Console and open the Amazon FinSpace console at https://console.aws.amazon.com/finspace.
2. Choose `finspace-saml-adfs` from the list of environments.
3. Copy the link under **Domain** and paste it in your web browser.

   You will be re-directed to your AD FS authentication page.
4. Enter your SSO credentials to sign in to FinSpace.

## Managing user access in Amazon FinSpace

Amazon FinSpace administrators or superusers can use the following topics to manage user access.

**Superuser**

A superuser has all the permissions in FinSpace. The first superuser for your FinSpace environment is created from the AWS console. The superuser can then create other superusers and application users from the FinSpace web application.

**Application user**

An application user does not have any permissions when their account is created. They are assigned permissions by adding them to a permission group.

**Permission group**

Permission groups contain users. Permissions to perform any action in FinSpace are assigned to permission groups, not directly to the user. A user can be a member of multiple permission groups. A permission group cannot be a member of another permission group.

**Permissions**

Permissions are assigned to permission groups and not to users. The are two kinds of permissions in FinSpace - application permissions and dataset permissions. Application permissions are assigned to a permission group when creating or editing it (for example, create datasets). Dataset permissions are assigned on a per dataset basis when associating a permission group to a dataset (for example, read a view in a dataset).

**Topics**



## Managing user access with email and password

**Important**

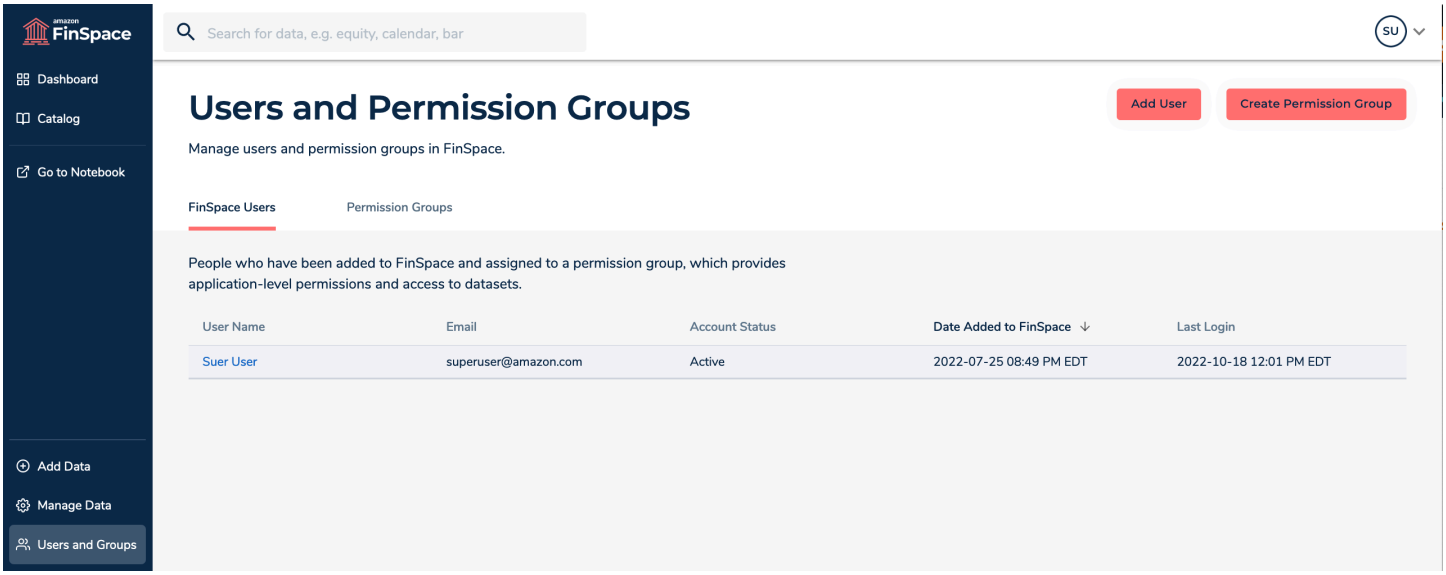
Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

This section describes how you can manage users in an Amazon FinSpace environment created with Email and password based authentication.

**Note**

To create and manage users, you must be a superuser or a member of a group with necessary permissions - **Manage Users and Permission Groups**.

You can invite users by creating an account for them and sharing access credentials.




### Creating the first superuser

The first superuser must be created after a new FinSpace environment is created. See details in this section. Once the first superuser is created, they can sign in to FinSpace web application and setup other superusers and application users. Subsequent superusers can be created by the first superuser in the FinSpace web application.

### Inviting users to access FinSpace

In FinSpace, you can invite users by creating an account for them and sharing access credentials. FinSpace accounts are created in two steps. First, you create a user in FinSpace. This creates an inactive account in FinSpace, credentials and a temporary password is generated for the user which is shared with them. When the user accepts the invitation and signs in for the first time, the user creates a new password to activate the account.

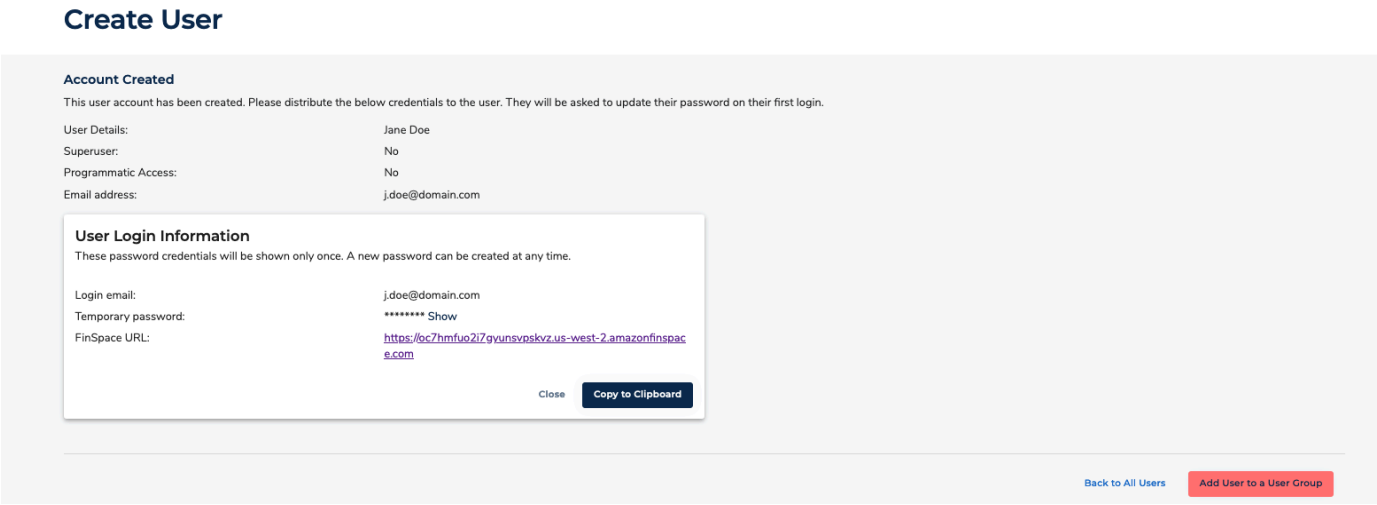
For more information about signing in for the first time, see Signing in to the Amazon FinSpace web application.

### To create accounts and invite users to FinSpace

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**.
3. On the **Users and Permission Groups** page, choose **Add User**.
4. On the **Create User** page, specify the **User Details**.

5. For **Superuser**, choose **Yes** to designate the user as a superuser or **No** to designate this user as an application user.
6. For **Programmatic Access**, choose **Yes** to provide access to use FinSpace APIs and SDK or choose **No** to deny programmatic access.

   When you choose **Yes**, you are required to specify the **IAM Principal ARN** for this user in the format `arn:partition:service::region::account::resource`.
7. Choose **Create User**.
8. After the account is created, copy the credentials to clipboard and share them with the new user.




## Viewing user details

### To view details of a user

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**. The **Users and Permission Groups** page, displays the list of users under the **FinSpace Users** tab.
3. Select a user to view their details.

### Deactivating a user

#### To deactivate a user

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**.
3. Choose **FinSpace Users** tab.
4. Select a user to view their details.
5. On the top right corner, choose **More** menu.
6. Choose **Deactivate User**. This button is only visible to superusers and users with with necessary permissions – **Manage Users and Permission Groups**.
7. On the confirmation dialog box, choose **Deactivate**. You can activate a user again later if necessary.

## Managing user access with SSO

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

This section describes how you can manage users in an Amazon FinSpace environment created with SAML based SSO authentication.

**Note**

1. In order to create and manage users, you must be a superuser or a member of a group with necessary permissions - **Manage Users and Permission Groups**.
2. You will need administrator privileges to assign and remove users to your configured FinSpace application in your Identity Provider.

You can invite users by creating a FinSpace account for them. When using SAML based Single Sign On as the authentication method for your FinSpace environment, you need to execute two steps to add users in FinSpace.

1. Assign user to your FinSpace application in your Identity Provider (IdP) with their email.
2. Create the user in FinSpace environment. The email of the user created in FinSpace environment must match their email in their identity record with the Identity provider.

If above steps are not followed, a user will not be successfully authenticated to use FinSpace.

**Creating the first superuser**

The first superuser must be created after a new FinSpace environment is created. The user must be assigned to the FinSpace application created in your IdP. See details in this section. Once the first superuser is created, they can sign in to FinSpace web application and setup other superusers and application users. Subsequent superusers can be created by the first superuser in the FinSpace web application.

**Inviting users to access FinSpace**

In FinSpace, you can invite users by creating a FinSpace account for them. For more information about signing in for the first time, see Signing in to the Amazon FinSpace web application.

**To create FinSpace accounts and invite users**

1. Assign the new user to the application created for FinSpace in your IdP.
2. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
3. On the left navigation bar of the home page, choose **Users and Groups**.
4. On the **Users and Permission Groups** page, choose **Add User**.
5. On the **Create User** page, specify the **User Details**. The email that you enter must match the email of the user record in your IdP.
6. For **Superuser**, choose **Yes** to designate the user as a superuser or **No** to designate this user as an application user.
7. For **Programmatic Access**, choose **Yes** to provide access to use FinSpace APIs and SDK or choose **No** to deny programmatic access.

When you choose **Yes**, you are required to specify the **IAM Principal ARN** for this user in the format `arn:partition:service::region::account::resource`.

8. Choose **Create User**.
9. After the account is created, copy the credentials to clipboard and share them with the new user. The user can sign in to FinSpace with their SSO credentials.

### Viewing user details

#### To view details of a user

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**. The **Users and Permission Groups** page, displays the list of users under the **FinSpace Users** tab.
3. Select a user to view their details.

### Deactivating a user

#### To deactivate a user

1. Remove the user from the list of assigned users from the FinSpace application in your Identity Provider (IdP).
2. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
3. On the left navigation bar of the home page, choose **Users and Groups**.
4. Choose **FinSpace Users** tab.
5. Select a user to view their details.
6. On the top right corner, choose **More** menu.
7. Choose **Deactivate User**. This button is only visible to superusers and users with necessary permissions – **Manage Users and Permission Groups**.
8. On the confirmation dialog box, choose **Deactivate**. You can activate a user again later if necessary.

# Managing user permissions with permission groups

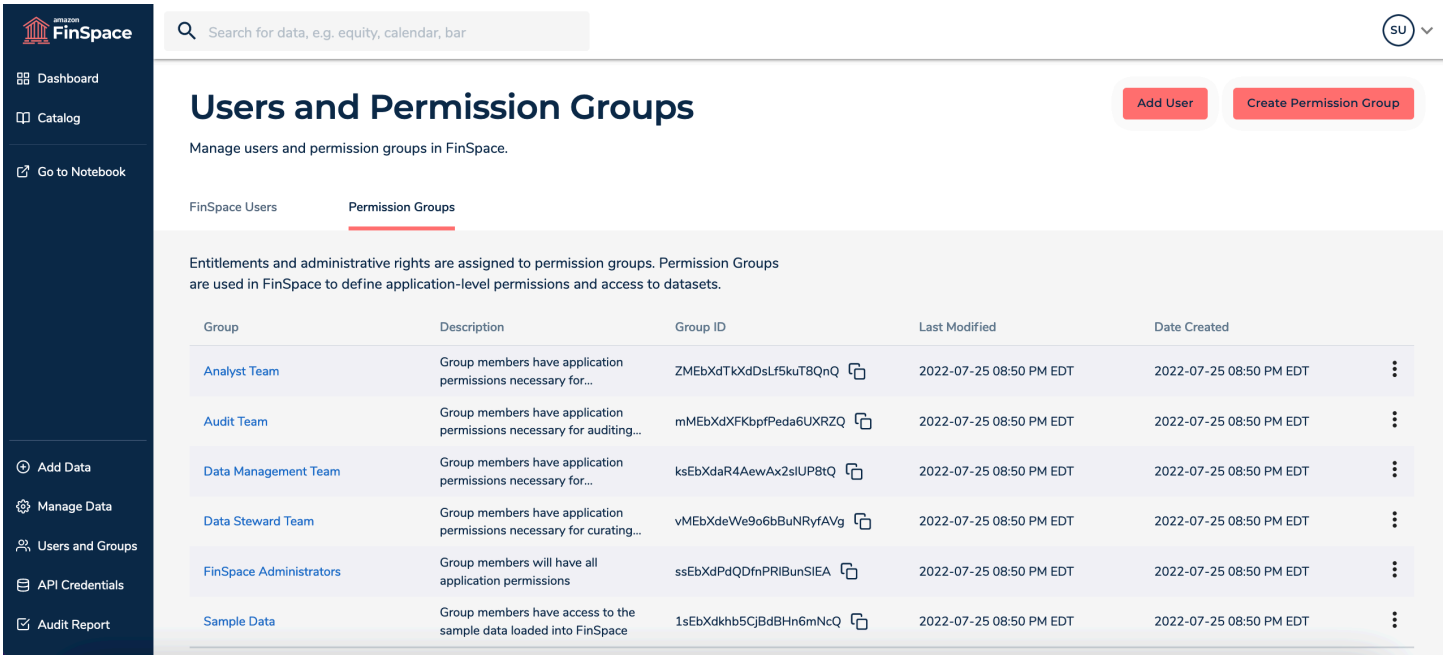
**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

**Note**

In order to create and manage permission groups, you must be a superuser or a member of a group with necessary permissions - **Manage Users and Permission Groups**.

You can create permission groups inside Amazon FinSpace, so you do not have manage permissions individually. Permissions are not assigned directly to a user but a permission group is created with the appropriate permissions, and a user is assigned to that permission group.



### Permissions

Permissions are assigned to permission groups and not to users. The are two kinds of permissions in FinSpace - application permissions and dataset permissions. Application permissions are assigned to a permission group when creating or editing it (for example, create datasets). Dataset permissions are assigned on a per dataset basis when associating a permission group to a dataset (for example, read a view in a dataset).

> **Warning**
>
> When assigning application permissions, be aware that the permission **Manage Users and Permission Groups** allows users to grant themselves or others access to any functionality in their FinSpace environment's application. It should only be granted to trusted users.

### Supported application permissions

| Permission | Description |
| --- | --- |
| Create Datasets | Group members can create new datasets in FinSpace or via the FinSpace API |
| Manage Categories and Controlled Vocabularies | Group members can create, edit and delete categories and controlled vocabularies |
| Manage Clusters | Group members will have permissions to manage clusters in FinSpace notebooks |
| Manage Users and Permission Groups | Group members can manage users and permission groups. This is a privileged permission that allows users to grant themselves or others access to any functionality in the application. It should only be granted to trusted users. |
| Manage Attribute Sets | Group members will have menu option to manage Attribute Sets |

| Permission | Description |
| --- | --- |
| Manage Attribute Sets | Group members can create, edit and delete attribute sets |
| View Audit Data | Group members can view audit data |
| Access Notebooks | Group members will have access to the FinSpace notebooks |
| Get Temporary Credentials | Group members will be able to get temporary API credentials |

**Supported dataset permissions**

When a dataset is created by a user, all other members of the same permission group will inherit access to the dataset. The members can permission the dataset to other permission groups and specify the actions that the other groups they can take on it. Users can only create a dataset if their permission group has application permission for **Create Datasets**.

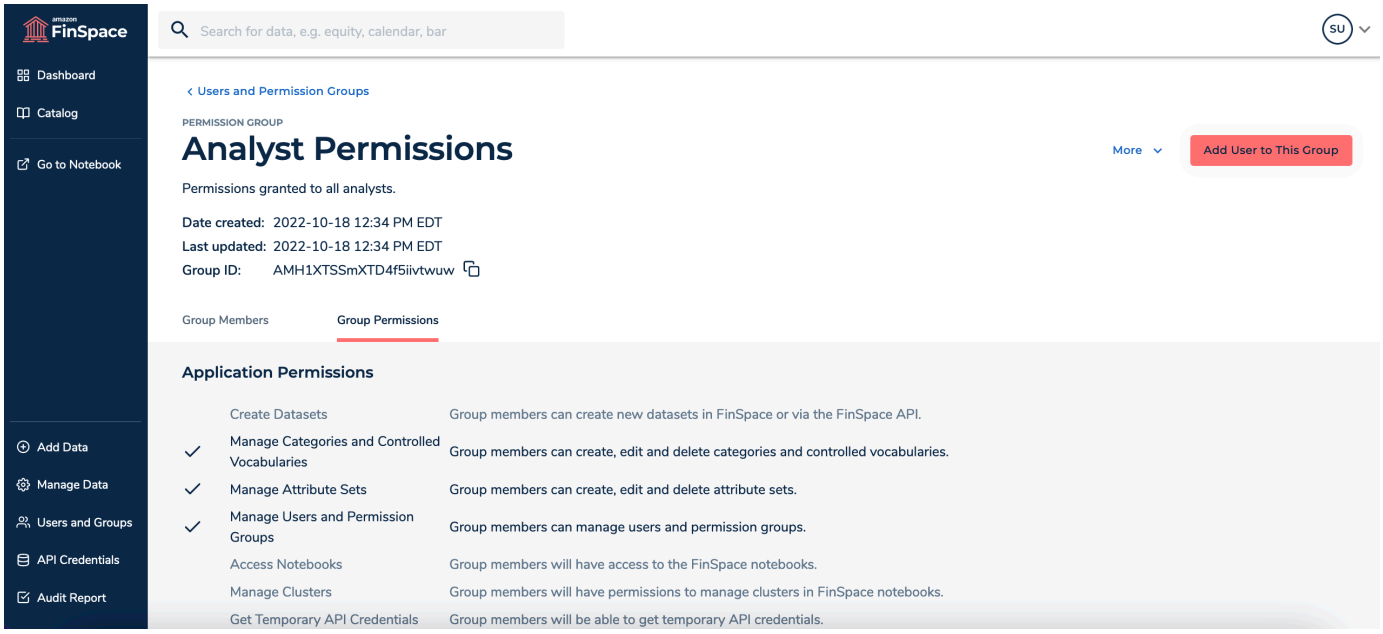
| Permission | Description |
| --- | --- |
| View Dataset Details | Group members can view dataset details |
| Read Dataset Data | Group members can read the data files, such as data views, provided on S3 for Spark, notebooks, and access from outside FinSpace |
| Add Dataset Data | Data Group members can add new data files to this dataset to create a dataset update |
| Create View | Group members can create new data or file view on this dataset via the Web UI or API |
| Edit Dataset Metadata | Group members will have permission to edit dataset metadata including permission to add additional attribute sets |

| Permission | Description |
| --- | --- |
| Manage Permissions | Group members can view and edit this dataset permissions |
| Delete Dataset | Group members can remove the dataset including all data and data views |

### Creating and adding a user to the group

### To create a permission group and add a new user to it

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**.
3. On the **Users and Permission Groups** page, choose **Create Permission Group**.
4. On the **Create Permission Group** page, enter the name and description for the permission group and select appropriate permissions for the group.
5. Choose **Create**. A new group is created with selected permissions.




6. Choose **Add User to This Group**.
7. On the dialog box, select a user to add to this group.
8. Choose **Add**. A new user is now added to the group.

### List all permission groups

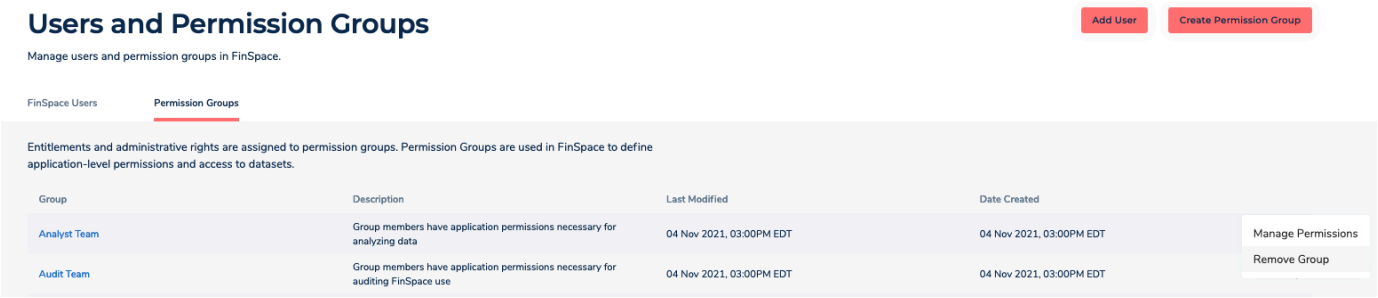
#### To list all created permission groups

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**.
3. Choose the **Permission Groups** tab. A list of all the permission groups is displayed in the table.

### Delete a permission group

#### To delete a permission group

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**.
3. Choose the **Permission Groups** tab.
4. From the list, select a group and choose the more (⋮) icon.
5. Choose **Remove Group**.




6. In the dialog box that appears, choose **Remove**.

## Temporary credentials in Amazon FinSpace

Amazon FinSpace has an internal application authorization model that controls access to the functions in FinSpace and the FinSpace API operations. In order to use the FinSpace API operations, you must first obtain temporary security credentials, which are used when you call these API

operations. These credentials are unique for each user and are only valid for 60 minutes. After the credentials expire, you need to obtain new credentials before making subsequent API calls.

### Obtaining the credentials using FinSpace

You can obtain credentials from the web application if you're one of the following:

- A superuser
- An application user who is a member of a FinSpace permission group with the **Get Temporary API Credentials** permission

### To obtain the permissions

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **API Credentials**.
3. On the **API Credentials** page, use the copy icon to copy the **Access Key ID**, **Secret Access Key**, and the **Session Token** values.
4. Use these copied credentials to access the FinSpace data API operations.

```
#!/usr/bin/env python

import boto3
session = boto3.session.Session()
finSpaceClient = session.client(
    region_name = 'us-east-1',
    service_name = 'finspace-data',
    aws_access_key_id = 'Specify Access Key ID',
    aws_secret_access_key = 'Specify Secret Access Key',
    aws_session_token = 'Specify Session Token'
)
```

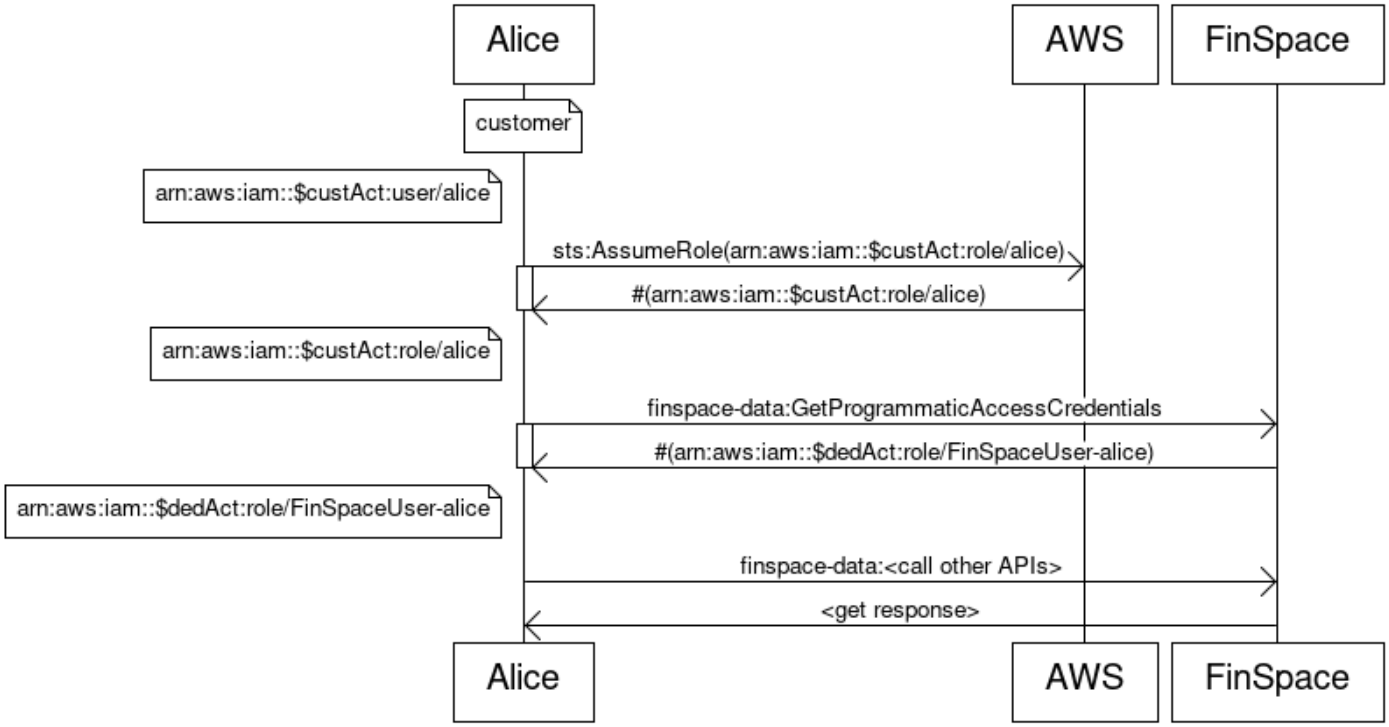
### Obtaining the credentials programmatically

You can also obtain the credentials using a program or a script without signing in to the FinSpace web application. For this, you can use the `GetProgrammaticAccessCredentials` API operation to retrieve the temporary credentials. You must call `GetProgrammaticAccessCredentials`

using the IAM role that exists in the AWS account that you used to create your Amazon FinSpace environment.

Calling the `GetProgrammaticAccessCredentials` API operation returns a set of temporary credentials that you can then use to call the other API operations. Before you obtain the temporary credentials, you need to enable the programmatic access for each user.

The following diagram illustrates how you can access and use the temporary credentials.




- The diagram shows that first a request to `AssumeRole` is sent to AWS. For more information, see AssumeRole in the *AWS Security Token Service API Reference*.
- This request returns a set of security credentials that are used to access the AWS resources.
- Next, a request is sent to `finspace-data` to call the `GetProgrammaticAccessCredentials` API operation. This request returns the temporary credentials.
- Lastly, the temporary credentials are used to call the other FinSpace API operations.

### Configuring a user for programmatic access using FinSpace

Use the following procedures to allow a specific user to obtain API credentials programatically.

> **Note**
>
> To perform the following steps, you must either be a superuser or a member of a group with necessary permissions – **Manage Users and Groups**.

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Users and Groups**.
3. On the **Users and Permission Groups** page, choose a user that you want to enable programmatic access for.
4. On the user details page, choose **More** and then choose **Edit User**.
5. For **Programmatic Access**, choose **Yes**.
6. For **IAM Principal ARN**, enter the ARN identifier for an IAM role that will be used. This role is used to call `GetProgrammaticAccessCredentials` to obtain temporary API credentials.

   The IAM role must reside in the AWS account that you used to create your FinSpace environment and must have the following permission set:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "finspace-api:GetProgrammaticAccessCredentials",
   "Resource": "arn:aws:finspace-api:<region>:<account-id>:/credentials/
programmatic"
  }
 ]
}
```

7. To save your edits to the user, choose **Update User**.

> **Note**
>
> Alternatively, you can also enable programmatic access for a user at the time when you create a user. For more information, see Adding users in FinSpace.

### Enabling programmatic access using the FinSpace API

You can also enable programmatic access for a user by using the `CreateUser` and `UpdateUser` API operations. The following are examples of how you can use the API operations.

### Example JSON for the `CreateUser` API operation

```
{
    "emailAddress": "testemail1@amazon.com",
    "type": "APP_USER",
    "firstName": "test",
    "lastName": "user",
    "apiAccess": "ENABLED",
    "apiAccessPrincipalArn": "arn:aws:iam::012345678910:role/TestRole"
}
```

### Example JSON for the `UpdateUser` API operation

```
{
    "type": "SUPER_USER",
    "firstName": "test",
    "lastName": "user",
    "apiAccess": "ENABLED",
    "apiAccessPrincipalArn": "arn:aws:iam::012345678910:role/TestRole"
}
```

## AWS managed policies for Amazon FinSpace

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed

policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

## AWS managed policy: AWSFinSpaceServiceRolePolicy

You can't attach AWSFinSpaceServiceRolePolicy to your IAM entities. This policy is attached to a service-linked role that allows FinSpace to perform actions on your behalf. For more information, see Using service-linked roles for FinSpace.

This policy grants FinSpace permissions to publish metrics.

### Permissions details

This policy includes the following permission.

- `cloudwatch` – Allows principals access to publish metrics to the AWS/FinSpace and AWS/Usage namespace in the AWS account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/FinSpace",
                        "AWS/Usage"
                    ]
                }
            },
            "Resource": "*"
```

```
        }
    ]
}
```

### FinSpace updates to AWS managed policies

View details about updates to AWS managed policies for FinSpace since this service began tracking these changes.

| Change | Description | Date |
| --- | --- | --- |
| AWSFinSpaceServiceRolePolicy – Updated policy | Updated the `AWSServiceRoleForFinSpace` policy to allow PutMetricData calls to AWS/Usage CloudWatch namespace. | November 17, 2023 |
| AWSFinSpaceServiceRolePolicy – New policy | FinSpace added a new policy to enable access to AWS service and resources used or managed by Amazon FinSpace. | June 5, 2023 |
| FinSpace started tracking changes | FinSpace started tracking changes for its AWS managed policies. | June 5, 2023 |

## Using service-linked roles for FinSpace

Amazon FinSpace uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to FinSpace. Service-linked roles are predefined by FinSpace and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up FinSpace easier because you don't have to manually add the necessary permissions. FinSpace defines the permissions of its service-linked roles, and unless

defined otherwise, only FinSpace can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your FinSpace resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

## Service-linked role permissions for FinSpace

FinSpace uses the service-linked role named **AWSServiceRoleForFinSpace** – Policy to enable access to AWS service and resources used or managed by Amazon FinSpace.

The AWSServiceRoleForFinSpace service-linked role trusts the following service to assume the role:

- `finspace.amazonaws.com`

The role permissions policy named AWSFinSpaceServiceRolePolicy allows FinSpace to complete the following action on the specified resources:

- Action: `cloudwatch:PutMetricData` on `*` in `AWS/FinSpace` and `AWS/Usage` CloudWatch namespace.

For more information about this policy, including the JSON policy document, see AWSFinSpaceServiceRolePolicy.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

## Creating a service-linked role for FinSpace

You don't need to manually create a service-linked role. When you create a FinSpace environment in the AWS Management Console, the AWS CLI, or the AWS API, FinSpace creates the service-linked role for you.

**Important**

This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role. Also, if you were using the FinSpace service before May 25, 2023, when it began supporting service-linked roles, then FinSpace created the AWSServiceRoleForFinSpace role in your account. To learn more, see A new role appeared in my IAM account.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create a FinSpace environment, FinSpace creates the service-linked role for you again.

## Editing a service-linked role for FinSpace

FinSpace does not allow you to edit the AWSServiceRoleForFinSpace service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

## Deleting a service-linked role for FinSpace

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

**Note**

If the FinSpace service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

**Note**

If you want to delete the AWSServiceRoleForFinSpace, you must first delete all of your FinSpace environments.

**To manually delete the service-linked role using IAM**

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForFinSpace service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

### Supported regions for FinSpace service-linked roles

FinSpace supports using service-linked roles in all of the regions where the service is available. For more information, see Regions and IP ranges.

## Data protection in Amazon FinSpace

The AWS shared responsibility model applies to data protection in Amazon FinSpace. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data privacy FAQ. For information about data protection in Europe, see the AWS shared responsibility model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use TLS to communicate with AWS resources. Clients must support TLS 1.2.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal information processing standard (FIPS) 140-2

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you

work with FinSpace or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into FinSpace or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

**Topics**



# Data encryption in Amazon FinSpace

**Amazon FinSpace uses the following data encryption features**



## Encryption at rest

To encrypt data at rest, Amazon FinSpace uses a customer-owned key from the AWS Key Management Service (AWS KMS). When you create a FinSpace environment, you can specify the KMS key that you want to use to encrypt all of the service data and metadata in your environment.

## Encryption in transit

Amazon FinSpace uses TLS 1.2 to encrypt data in transit.

# Inter-network traffic privacy in Amazon FinSpace Dataset browser

**Take following network considerations into account when using the Amazon FinSpace web application**

1. To use FinSpace web application, you need access to the internet.
2. You will need access to a compatible browser.
3. Your connections to FinSpace are protected through the use of TLS. So that you can access the FinSpace notebook environment that runs on SageMaker Studio, you must allow access to HTTPS and WebSockets Secure (wss://) protocol. You will need to allow-list access to SageMaker to access the Notebook environment. An example for allow-listing string is `*.us-`

`east-1.sagemaker.aws`. You may change the region depending on the region you have setup FinSpace.

4. By default, FinSpace notebooks allow public internet access. You can request the access be blocked by contacting AWS support.

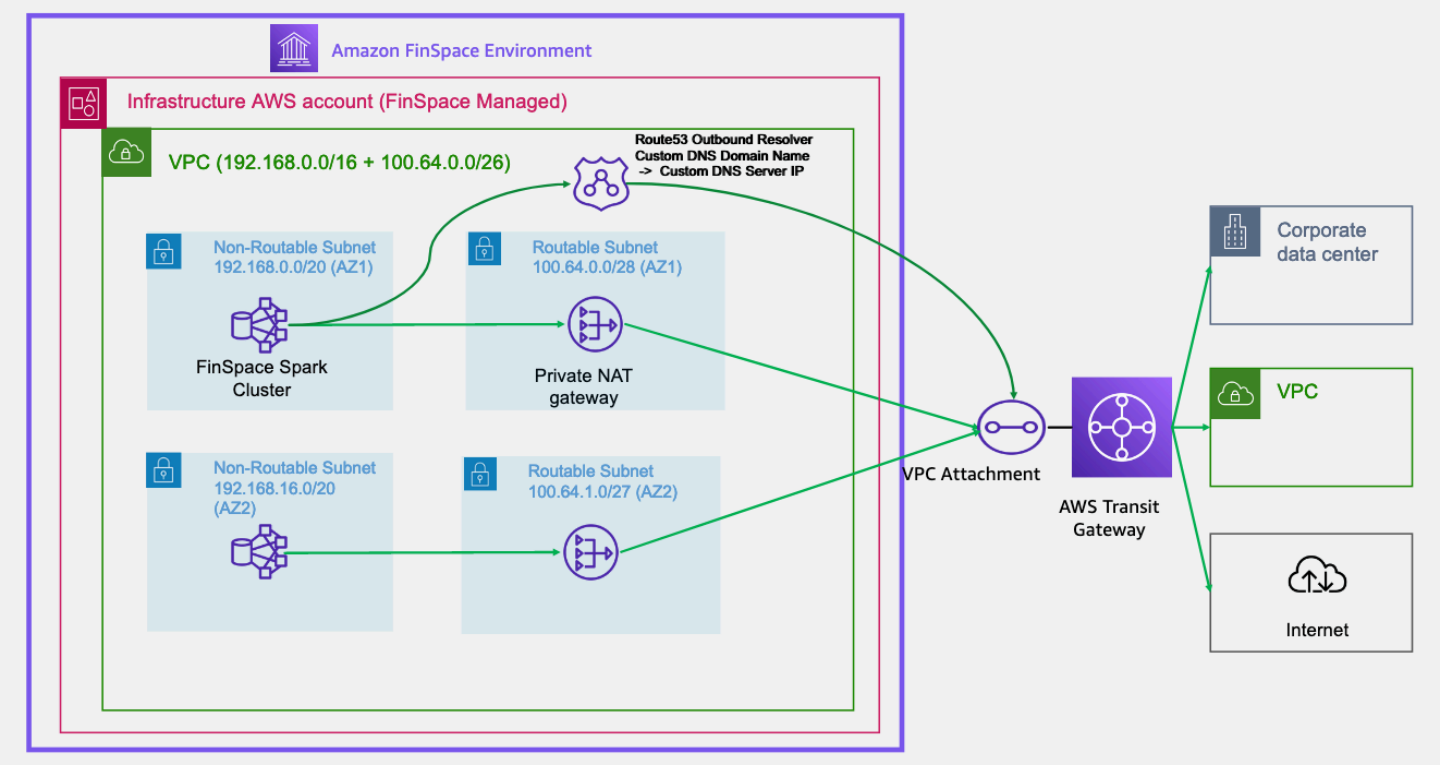
# Connecting Amazon FinSpace to your network

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can use a FinSpace virtual private cloud (VPC) connection to allow the compute resources running in your FinSpace environment infrastructure account to access resources in your internal network. For example, a FinSpace analyst can connect their FinSpace managed Spark cluster to an internal code repository or internal application REST service. Using this feature, you can also connect to databases on your corporate network and access data, which you could then combine with data in FinSpace.

## How a FinSpace VPC connection works

You create a FinSpace VPC connection by connecting your FinSpace infrastructure account to an existing transit gateway in your organization. You can configure the transit gateway to route traffic to other portions of your network. The following diagram shows how a FinSpace VPC connection works.




The diagram describes a high-level setup of a FinSpace VPC connection:

- Each FinSpace environment contains a dedicated, service-managed AWS account called an environment infrastructure account.
- In this account, there is a VPC. This VPC contains non-routable subnets that host FinSpace managed compute resources. The VPC also contains routable subnets that host a private NAT gateway. The private NAT gateway is connected to a customer managed transit gateway through a transit gateway attachment.
- As shown in the diagram, you can connect the transit gateway that you manage to additional parts of your network, including your VPCs and on-premises networks. You can also configure a network path from this transit gateway to the internet if you want to.
- The non-routable subnets that are in the VPC in the FinSpace infrastructure account use ranges within a Classless Inter-Domain Routing (CIDR) block of *192.168.0.0/16*. The routable subnets use CIDR ranges that you provide. This diagram shows an example of using a *100.64.0.0/26* CIDR range, which you provide for use in two Availability Zones (AZs).
- In the VPC of the environment infrastructure account, a Route 53 outbound resolver forwards custom DNS queries to a custom DNS server that you specify.
- FinSpace creates multiple AZs in a Region and private NAT gateways in every AZ.

### DNS resolution

The VPC in the environment infrastructure account contains a Route 53 resolver that is used by the hosts for DNS lookups. By default, after you configure a VPC connection, this resolver resolves AWS service names, but not other hosts on your network or the internet.

When you set up your FinSpace VPC connection, you can optionally configure this resolver so that it forwards queries to a resolver that you specify. This allows hosts that are running in the FinSpace infrastructure account to be able to resolve hostnames from this resolver.

**Topics**



## Managing a FinSpace VPC connection

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

This section explains how to set up and remove a FinSpace virtual private cloud (VPC) connection.

### Prerequisites

Before you proceed, complete the following prerequisites:

- Make sure that a FinSpace environment has been created. For more information, see Setting up an Amazon FinSpace environment.
- Make sure that a transit gateway has been created in AWS Transit Gateway. For more information, see Create the transit gateway in the *AWS Transit Gateway User Guide*.
- Make sure that you've gathered the following information to create an AWS Support case to request access:

- FinSpace environment ID.
- AWS Region of the FinSpace environment.
- Transit gateway ID of the transit gateway that you will connect your FinSpace environment to.
- The IP address range to use for the customer-facing side of the NAT gateway. This should be a /26 IP address range from the 100.64.0.0/10 range that is specified by RFC 6598.
- (Optional) Custom DNS domain name – The name of the domain for which the DNS queries are forwarded to custom DNS server IP address.
- (Optional) Custom DNS server IP address – The IP address that's routable from your transit gateway attachment.

## Considerations

Before you get started with the setup, make sure that you review the following considerations:

- The /26 IP address range for the routable subnets that is attached to the transit gateway must be from the 100.64.0.0/10 range specified by RFC 6598.
- The /26 IP address range for the routable subnets that is attached to the transit gateway must be unique across FinSpace environments and your network that's connected to the same transit gateway. For example, you might have two FinSpace environments (environment-A and environment-B) that are connected to TGW-A. Ensure that the /26 CIDR provided for each environment is distinct across environment-A and environment-B, and your network connected to the TGW-A.

## Setting up a VPC connection

### To set up a VPC connection

1. Sign in to the AWS Support Center Console.
2. Open a technical support case to enable the VPC connection for FinSpace, and provide the following information:
   - The FinSpace environment ID
   - The transit gateway ID
   - The AWS Region of the FinSpace environment
   - The /26 IP range to use for the customer-facing side of the NAT gateway

- (Optional) The custom DNS domain name
- (Optional) The custom DNS server IP address

For more information, see Creating a support case in the *AWS Support User Guide*.

3. Create a RAM share for your transit gateway to the FinSpace environment infrastructure account. For more information, see Share a transit gateway in the *AWS Transit Gateway User Guide*.
4. After verifying the support case, a FinSpace operator runs a setup program. This program accepts the RAM share request, disables internet in the FinSpace environment infrastructure account, and issues a VPC attachment request to your transit gateway.
5. When the request is complete, the FinSpace operator sends a notification, and adds the transit gateway attachment ID and the Availability Zone (AZ) to the VPC attachment request.
6. Accept the VPC attachment request that FinSpace issues to your transit gateway. For more information, see Accept a shared attachment in the *AWS Transit Gateway User Guide*.
7. Configure the routing tables in your transit gateway traffic, and route to/from the subnets in the VPC that were attached in the VPC attachment.

> **Note**
>
> Ensure that your transit gateway attachment is created with all the Availability Zones provided in the notification that you receive from the FinSpace operator.

8. Ensure that the VPC connection setup is successful by following the steps in Validating your VPC connection.

## Removing a VPC connection

### To remove an existing VPC connection

1. Delete the transit gateway attachment from your transit gateway. For more information, see Delete a VPC attachment in the *AWS Transit Gateway User Guide*.
2. After removing the attachment, restore direct internet access to your FinSpace environment by creating a new technical support case that specifies the environment ID.

**Note**

Deleting a FinSpace environment does not automatically delete the attachment. You must remove the attachment separately.

### Updating a VPC connection

You cannot update an existing connection. To make changes to an existing connection, remove the old connection and create a new one.

## Validating your VPC connection

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

After your connection is set up, you can test connectivity to your network. There are two types of connectivity testing available.

- Basic testing – You can use the `curl` command in the SageMaker Studio notebook environment that is included with FinSpace for basic testing.
- Advanced testing – You can use the file upload capability in the SageMaker Studio notebook to upload your own network diagnostic utilities to test.

### Basic connectivity testing using Amazon FinSpace notebooks

If you're testing connectivity to an HTTP/HTTPS endpoint, you can use a FinSpace notebook to test basic connectivity using curl.

**To validate the connection using a notebook**

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. Open a FinSpace notebook. For more information, see Opening the notebook environment.
3. From the notebook menu bar, choose the plus (+) icon to create a new cell. In the cell, run the `curl` command in the shell to test connectivity to the host.

   ```
   %local

   !curl <hostname or URL>
   ```

   For example, run the following command:

   ```
   %local

   !curl www.google.com
   ```

   > **Tip**
   >
   > Keep your cursor in the cell and choose the (>) arrow button from the notebook menu to run the command.

   If successful, the results of the `curl` command display in your notebook.

> **Note**
>
> After the VPC connectivity is set up for the environment, the internet connection is disabled by default. This is the default unless you have an explicit static route entry in your transit gateway route table that specifies forwarding all traffic `0.0.0.0/0` to your VPC attachment that has an internet gateway.

### Configuring internet access

**To ensure FinSpace can connect to your internet gateway**

1. Check that your transit gateway attachment has a private subnet and public subnet with a NAT gateway. The private subnet should be attached to the transit gateway attachment.
2. Check that the VPC attachment has all Availability Zones (AZ) included in the FinSpace response.
3. Add a route for private subnets for directing traffic destined to `100.64.0.0/26` to the dedicated account VPC attachment.
4. Create a transit gateway static route to direct traffic destined to `0.0.0.0/0` to the customer account attachment. For more information, see Create a static route in the *AWS Transit Gateway User Guide*.

   Wait a few minutes before running the next command because there might be a delay before the routes are installed.

## Monitoring IP traffic

You can use the transit gateway flow logs to monitor traffic coming from FinSpace. For more information, see Logging network traffic using Transit Gateway Flow Logs in the *AWS Transit Gateway User Guide*.

# Resilience in Amazon FinSpace

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

While FinSpace is multi-AZ, it does not support backups to other AWS Availability Zones or Regions. However, you can write your own application using the FinSpace SDK to query data and save it to the destination of your choice.

# Infrastructure security in Amazon FinSpace

As a managed service, Amazon FinSpace is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access FinSpace through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

FinSpace is architected so that your traffic is isolated to the specific AWS Region that your FinSpace environment resides in.

## Connect to FinSpace using an interface VPC endpoint

You can connect to FinSpace APIs using an interface VPC endpoint (AWSPrivateLink) instead of connecting over the internet. When you use an interface VPC endpoint, communication between your VPC and FinSpace is conducted entirely within the AWS network. Each VPC endpoint is represented by one or more Elastic network interfaces (ENIs) with private IP addresses in your VPC subnets.

> **Note**
>
> You can only connect to FinSpace web application over the internet.

To use FinSpace through your VPC, you must connect from an instance that is inside the VPC or connect your private network to your VPC by using an Amazon Virtual Private Network (VPN) or AWS Direct Connect. For information about Amazon VPN, see VPN connections in the Amazon

Virtual Private Cloud User Guide. For information about AWS Direct Connect, see Creating a connection in the AWS Direct Connect User Guide.

FinSpace supports VPC endpoints in all AWS Regions where both Amazon VPC and FinSpace are available.

You can create an interface VPC endpoint to connect to FinSpace using the AWS console or AWS Command Line Interface (AWS CLI) commands. For more information, see Creating an interface endpoint.

You will need to create separate endpoints for using FinSpace management APIs and Data APIs:

- Management APIs – `com.amazonaws.<Region>.finspace`
- Data APIs – `com.amazonaws.<Region>.finspace-api`

After you create an interface VPC endpoint, if you enable private DNS hostnames for the endpoint, the default FinSpace endpoint resolves to your VPC endpoint.

For more information, see Interface VPC endpoints (AWS PrivateLink) in the Amazon VPC User Guide.

### Create a VPC endpoint policy for FinSpace

You can create a policy for Amazon VPC endpoints for FinSpace to specify the following:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the Amazon VPC User Guide. Whenever you use IAM policies, make sure that you follow IAM best practices. For more information, see Security best practices in IAM in the AWS Identity and Access Management User Guide.

## Security best practices in Amazon FinSpace

Amazon FinSpace provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't

represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

- Implement least privilege access.
- Limit access to sensitive and important auditing functions.
- When creating resources through the update or bulk import APIs, do not use PHI or PII, including the names of datastores and jobs, in any visible fields.

# Querying AWS CloudTrail logs

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or AWS service in FinSpace. CloudTrail captures all API calls for FinSpace as events. The events captured include calls from the FinSpace console, web application, and code calls to the FinSpace APIs. You can use the information collected by CloudTrail to determine the request that was made to FinSpace, the IP address of the requester, who made the request, when the request was made, and additional details.

You can create a trail to enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket, including events for FinSpace. If you don't configure a trail, you can still view the most recent events in the CloudTrail console.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

## FinSpace information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. In the CloudTrail console in **Event history**, you can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail event history in the AWS CloudTrail User Guide.

For an ongoing record of events in your AWS account, including events for FinSpace, create a trail. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following in the *AWS CloudTrail User Guide*:

- Creating a trail for your AWS account
- AWS service integrations with CloudTrail logs
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions
- Receiving CloudTrail log files from multiple accounts

CloudTrail logs all FinSpace API operations including actions taken in the FinSpace web application. These and other operations are documented in the API references:

- Amazon FinSpace management API reference
- Amazon FinSpace data API reference

Every event or log entry contains information about who generated the request. The identity information helps you determine:

- The details of the user that made the request.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see CloudTrail userIdentity element in the AWS CloudTrail User Guide.

## Understanding FinSpace log file entries

CloudTrail delivers events as log files that contain one or more log entries. An event represents a single request from any source and includes information about the requested operation, the date and time of the operation, the request parameters, and so on. Because these log files aren't an ordered stack trace of the public API calls, they don't appear in any specific order.

The following example CloudTrail log entry demonstrates the `CreateEnvironment` operation, which creates a new FinSpace environment.


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AROARFVIKXOEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "AROARFVIKXOEXAMPLE",
    "userName": "johndoe"
  },
  "eventTime": "2021-03-16T17:14:44Z",
  "eventSource": "finspace.amazonaws.com",
  "eventName": "CreateEnvironment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.197.99",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101
 Firefox/78.0",
  "requestParameters": {
    "name": "TestEnv",
    "federationMode": "LOCAL",
    "kmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/d9610405-8674-450f-b8cd-
e47999fdf2d",
    "tags": {}
  },
  "responseElements": {
    "environmentArn": "arn:aws:finspace:us-
east-1:123456789012:environment/6c6b4bbnnxin774ruft2dr",
    "environmentId": "6c6b4bbnnxin774ruft2dr",
    "environmentUrl": "6c6b4bbnnxin774ruft2dr.us-east-1.amazonfinspace.com"
  },
  "requestID": "167148e31-951f-52a8-b9bd-be347ce7801f",
  "eventID": "c2949aca-8862-4903-970e-64ae7cc1ba6b",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "120345661733"
}
```

The following example CloudTrail log entry demonstrates the `GetEnvironment` operation, which describes a FinSpace environment.


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AROARFVIKXOEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "AROARFVIKXOEXAMPLE",
    "userName": "johndoe"
  },
  "eventTime": "2021-02-26T13:59:00Z",
  "eventSource": "finspace.amazonaws.com",
  "eventName": "GetEnvironment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.22.198.64",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:85.0) Gecko/20100101
 Firefox/85.0",
  "requestParameters": {
    "environmentId": "ks56piapqiwaqxwj4xsjxx"
  },
  "responseElements": null,
  "requestID": "94ac7fff-1aad-4470-b0d9-83d13432ae4b",
  "eventID": "d318ca50-c45e-4f2d-954e-d23bee29effa",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012"
}
```


The following example CloudTrail log entry demonstrates the `CreateUser` operation, which creates a FinSpace user by using the FinSpace data API.

In this example, the value of the `principalId` element is the FinSpace user ID of the user who accesses the web application.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
```

```
        "principalId": "gmkd6xsrn9h7hfgxtnqlxw",
        "accountId": "123456789012"
    },
    "eventTime": "2022-12-12T19:57:48Z",
    "eventSource": "finspace-api.amazonaws.com",
    "eventName": "CreateUser",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "52.94.133.129",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:85.0) Gecko/20100101
    Firefox / 85.0 ",
    "requestParameters": {
        "emailAddress": "*** REDACTED ***",
        "firstName": "*** REDACTED ***",
        "lastName": "*** REDACTED ***",
        "type": "APP_USER"
    },
    "responseElements": {
        "userId": "tskd9r67fvb6yxtmda6wla"
    },
    "additionalEventData": {
        "finspaceDisplayableOperationName": "Create user",
        "finspaceEnvironmentId": "nlbapur76fhbij6oohyfyu"
    },
    "requestID": "771abcc3-0539-414e-9310-ec123eaa6d01",
    "eventID": "390f33ab-4396-4f18-8126-599016fe7280",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "apiVersion": "2020-07-13",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

The following example CloudTrail log entry demonstrates the `CreateDataset` event generated by an action taken in the FinSpace web application.

In this example, the value of the `principalId` element is the FinSpace user ID of the user who accesses the web application.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
```

```
        "principalId": "gmkd6xsrn9h7hfgxtnqlxw",
        "accountId": "123456789012"
    },
    "eventTime": "2022-12-12T20:06:46Z",
    "eventSource": "finspace-api.amazonaws.com",
    "eventName": "CreateDataset",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "52.94.133.129",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X     10 _15_7) AppleWebKit /
537.36(KHTML, like Gecko) Chrome / 107.0 .0 .0 Safari / 537.36 ",
    "requestParameters": {
        "datasetDescription": "*** REDACTED ***",
        "datasetTitle": "Test Dataset",
        "kind": "TABULAR",
        "ownerInfo": {
            "name": "*** REDACTED ***",
            "email": "*** REDACTED ***"
        },
        "permissionGroupParams": {
            "permissionGroupId": "4MIH3qXyXX9aRhDpRIFRQg",
            "datasetPermissions": [{
                "permission": "ViewDatasetDetails"
            }, {
                "permission": "ReadDatasetData"
            }]
        },
        "schemaDefinition": {
            "tabularSchemaConfig": {
                "columns": [{
                    "dataType": "DATETIME",
                    "name": "timestamp",
                    "description": "*** REDACTED ***"
                }, {
                    "dataType": "STRING",
                    "name": "event_type",
                    "description": "*** REDACTED ***"
                }]
            }
        }
    },
    "responseElements": {
        "datasetId": "b3m7g70"
    },
    "additionalEventData": {
```

```
        "finspaceEnvironmentId": "nlbapur76fhbij6oohyfyu"
    },
    "requestID": "f74eda06-3937-4b74-aea7-3c1ae176b82f",
    "eventID": "5fb6113e-c3c9-4dd6-a50f-0500b8b90d5b",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "apiVersion": "2020-07-13",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

## FinSpace data plane events in CloudTrail

Data access logging helps to log actions taken against the managed S3 bucket in your Amazon FinSpace environment's infrastructure account. The logging includes activity originating from the FinSpace managed Apache Spark clusters, FinSpace managed SageMaker Studio Notebook, and the FinSpace service itself. To enable logging of data access actions in FinSpace, you need to enable logging of data activity in CloudTrail. For more information, see Logging data events for trails.

The following example CloudTrail data event log entry demonstrates the event generated by accessing a dataset from Amazon S3 CLI.

The `principalId` element contains details about the FinSpace user ID that accesses a given FinSpace dataset ID. In this example, the user ID is the string ***jmiupn9hiyavwdw6pwdyva*** with the prefix ***u_*** and the dataset ID is the string ***64hzb00*** with the prefix ***ds_***.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId":
 "AROA6J3NRQLZNQALLTUBQ:S3Read_u_jmiupn9hiyavwdw6pwdyva_ds_64hzb00",
        "accountId": "123456789012"
    },
    "eventTime": "2022-12-12T00:44:59Z",
    "eventSource": "finspace-api.amazonaws.com",
    "eventName": "GetObject",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "3.142.242.209",
```

```
    "userAgent": "[aws-internal/3 aws-sdk-java/1.12.348
 Linux/4.14.296-222.539.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/25.352-b09 java/1.8.0_352
 vendor/Oracle_Corporation cfg/retry-mode/standard exec-env/AWS_ECS_FARGATE]",
    "requestParameters": {
        "bucketName": "finspace-us-east-2-hlbc5his5h6bgr3qgfstui",
        "Host": "finspace-us-east-2-hlbc5his5h6bgr3qgfstui.s3.us-east-2.amazonaws.com",
        "key": "ds/64hzb00/sn/gl/
Global_Market_Holidays_and_Timings_autoupdate_QMIuqNmW6nAYEKDGTRXJAw/
MMIuqQ20Kwku5t4fQuejUA/part-00000-4eff3050-b774-454b-9013-425840fb8057-
c000.snappy.parquet"
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "SigV4",
        "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "bytesTransferredIn": 0,
        "AuthenticationMethod": "AuthHeader",
        "x-amz-id-2": "7vrvbMFD5sIc/sdh9FHH2xxW5cY9ANw6J86lcszadwPaWusUAnhby2a45Hdw/
yPlUyOBJSJvvq0=",
        "bytesTransferredOut": 167507
    },
    "requestID": "EFJT8ER65E6T17N7",
    "eventID": "675b77c4-fc8c-4083-8716-0e137f374848",
    "readOnly": true,
    "resources": [{
        "accountId": "123456789012",
        "type": "AWS::FinSpace::Environment",
        "ARN": "arn:aws:finspace:us-east-2:123456789012:environment/
hlbc5his5h6bgr3qgfstui"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "sharedEventID": "17bd447a-d738-4cab-8fd8-f820e2cf179b",
    "eventCategory": "Data"
}
```

# Generating dataset browser audit report in Amazon FinSpace

**Important**

Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

You can generate audit reports to support your governance processes right from the FinSpace dataset browser by using the web application. FinSpace captures all activity within a FinSpace environment.

**Note**

- The audit report functionality only applies to FinSpace dataset browser. It does not apply for activity in Managed kdb Insights.
- In order to generate an audit report, you must be a superuser or a member of a group with necessary permissions - **View Audit Data**.

**Use the following procedure to generate an audit report**

1. Sign in to the FinSpace web application. For more information, see Signing in to the Amazon FinSpace web application.
2. On the left navigation bar of the home page, choose **Audit Report**.
3. On the **Generate Audit Report** page, choose one or more activity type.
4. Choose the period over which the report should be run.
5. (Optional) Filter the report by a user by specifying their email.
6. (Optional) Specify Dataset ID to filter the activity by a specific dataset.
7. Choose **RUN REPORT**.
8. (Optional) Export the audit report to Comma-separated values (CSV) file by choosing **DOWNLOAD FULL REPORT (.CSV)**.

# Definitions of columns in the audit report

| Audit report column | Description |
| --- | --- |
| Timestamp | The date and time of the event |
| Event Type | Type of the event. For example - user login, user accessing data content |
| Event | Details of the event |
| User | Email of the user related to the audit activity |
| Dataset ID | Dataset ID related to the event when applicable |

# Event types

| Event type | Description |
| --- | --- |
| Authentication | Events related to user sign in or accessing temporary credentials to use the API |
| Dataset Content | Events related to accessing and using a dataset |
| Dataset Definition | Events related to associating and updating an attribute set to a dataset |
| Categories | Events related to creating, editing, and removing categories |
| Attribute Sets | Events related to creating, editing, and removing attribute sets |
| Users and Permissions | Events related to creating, editing, and removing users and permission group permissions |

| Event type | Description |
| --- | --- |
| Spark Clusters | Events related to creating, scaling, and terminating spark clusters |
| Notebooks | Events related to creating, modifying, and terminating notebooks |
| Search | Events related to searching for datasets or browsing for datasets via data browser |
| Audit | Events related to generating, viewing, and downloading audit reports |

# Amazon FinSpace service quotas

> **Important**
>
> Amazon FinSpace Dataset Browser will be discontinued on *November 29, 2024*. Starting *November 29, 2023*, FinSpace will no longer accept the creation of new Dataset Browser environments. Customers using Amazon FinSpace with Managed Kdb Insights will not be affected. For more information, review the FAQ or contact AWS Support to assist with your transition.

Amazon FinSpace provides different resources that you can use. These resources include resources like environments, databases, volumes, clusters, scaling groups, etc. When you create your AWS account, we set default quotas on these resources on a per-Region basis.

The Service Quotas is a central location where you can view and manage your quotas for AWS services, and request a quota increase for many of the resources that you use. Use the quota information that we provide to manage your AWS infrastructure. Plan to request any quota increases in advance of the time that you'll need them.

You can contact AWS Support to request a quota increase for the service quotas listed in the AWS General Reference.

## View your current quotas

You can view your quotas for each Region using Service Quotas console.

**To view your current quotas using the Service Quotas console**

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/home/services/finspace/quotas/.
2. From the navigation bar (at the top of the screen), select a Region.


3. Use the filter field to filter the list by resource name. For example, enter **`kx.s.xlarge nodes`** to locate the quotas for these nodes.
4. To view more information, choose the quota name to open the details page for the quota.

## Request an increase

You can request a quota increase for each Region.

**To request an increase using the Service Quotas console**

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/home/services/finspace/quotas/.
2. From the navigation bar (at the top of the screen), select a Region.

3. Use the filter field to filter the list by resource name. For example, enter **`kx.s.xlarge nodes`** to locate the quotas for these nodes.
4. If the quota is adjustable, choose the quota and then choose **Request quota increase**.
5. For **Increase quota value**, enter the new quota value.
6. Choose **Request**.
7. To view any pending or recently resolved requests in the console, choose **Dashboard** from the navigation pane. For pending requests, choose the status of the request to open the request receipt. The initial status of a request is **Pending**. After the status changes to **Quota requested**, you'll see the case number with AWS Support. Choose the case number to open the ticket for your request.

# Quotas

The following table describes throttling quotas for application and user management within FinSpace.

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Environments | 2 | Yes | The maximum number of FinSpace environments you can create per AWS account |
| Users | 5 | Yes | The maximum number of users that can exist in a FinSpace environment |
| Permission groups | 20 | Yes | The maximum number of permission groups per FinSpace environment |

The following table describes the quotas for data within FinSpace.

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Datasets | 1500 | Yes | The maximum number of datasets that can exist in a FinSpace environment. |

| Name | Quota | Adjustable | Description |
| --- | --- | --- | --- |
| Concurrent changesets processing | 10 | Yes | The maximum number of concurrent changesets that can be processing per FinSpace environment. |
| Files per changeset | 100000 | No | The maximum number of files in a single changeset. |
| File size per changeset | 50 Gigabytes | No | The maximum file size of any single file in a changeset. |
| Data views per dataset | 3 | Yes | The maximum number of data views that can be created per dataset. |
| Concurrent data views processing | 10 | Yes | The maximum number of concurrently running data views processing per FinSpace environment. |
| Controlled vocabularies and categories | 100 | Yes | The maximum combined number of controlled vocabularies and categories per FinSpace environment. |
| Attribute sets | 100 | Yes | The maximum number of attribute sets that can exist in a FinSpace environment. |
| Datasets per permission group | 1500 | Yes | The maximum number of datasets assigned per permission group. |
| Notebook storage | 10 Gigabytes | No | The maximum amount of EFS storage per user notebook environment. |

The following table describes the quotas for compute within FinSpace.

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Clusters per user | 1 | No | The maximum number of FinSpace Spark clusters that can be active for each user. |

The following table describes the quotas for FinSpace Managed kdb resources.

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Managed kdb Multi-AZ clusters | 1 | Yes | The maximum number of Multi-AZ clusters per environment. |
| Managed kdb Single-AZ clusters | 5 | Yes | The maximum number of Single-AZ clusters per environment. |
| Managed kdb cluster users | 1,000 | Yes | The maximum number of cluster users per environment. |
| Managed kdb clusters | 10 | Yes | The maximum number of clusters allowed per environment. |
| Managed kdb concurrent changeset ingestions | 10 | Yes | The maximum number of concurrent changeset ingestions allowed per environment. |
| Managed kdb database cluster cache size | 7,730 Gigabytes | Yes | The maximum amount of database cluster cache allowed per environment. |

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Managed kdb databases | 1,500 | Yes | The maximum number of databases allowed per environment. |
| Managed kdb nodes per cluster | 5 | Yes | The maximum number of nodes per cluster. |
| Managed kdb savedown storage | 17,179 Gigabytes | yes | The maximum amount of savedown storage allowed per environment. |
| Total kdb environments | 1 | Yes | The maximum number of Managed kdb environments per AWS Account. |
| kx.s.16xlarge nodes | 0 | Yes | The maximum number of kx.s.16xlarge nodes allowed per environment. |
| kx.s.2xlarge nodes | 5 | Yes | The maximum number of kx.s.2xlarge nodes allowed per environment. |
| kx.s.32xlarge nodes | 0 | Yes | The maximum number of kx.s.32xlarge nodes allowed per environment. |
| kx.s.4xlarge nodes | 1 | Yes | The maximum number of kx.s.4xlarge nodes allowed per environment. |
| kx.s.8xlarge nodes | 1 | Yes | The maximum number of kx.s.8xlarge nodes allowed per environment. |
| kx.s.large nodes | 5 | Yes | The maximum number of kx.s.large nodes allowed per environment. |
| kx.s.xlarge nodes | 5 | Yes | The maximum number of kx.s.xlarge nodes allowed per environment. |

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Managed kdb changeset files | 262144 | No | The maximum number of files per changeset. |
| Managed kdb changeset single file size | 1 Terabyte | No | The maximum size of a single file in changesets. |
| Managed kdb changeset total size | 5 Terabytes | No | The maximum limit for total file size per changeset. |

The following table describes the quotas for Managed kdb scaling groups and volumes within FinSpace.

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Managed kdb scaling groups | 10 | Yes | The maximum number of Managed kdb scaling groups per environment. |
| kx.sg.large | 1 | Yes | The maximum number of kx.sg.large Managed kdb scaling group nodes per environment. |
| kx.sg.xlarge group nodes | 1 | Yes | The maximum number of kx.sg.xlarge Managed kdb scaling group nodes per environment. |
| kx.sg.2xlarge group nodes | 1 | Yes | The maximum number of kx.sg.2xlarge Managed kdb scaling group nodes per environment. |

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| kx.sg.4xlarge scaling group nodes | 1 | Yes | The maximum number of kx.sg.4xlarge Managed kdb scaling group nodes per environment. |
| kx.sg.8xlarge scaling group nodes | 1 | Yes | The maximum number of kx.sg.8xlarge Managed kdb scaling group nodes per environment: |
| kx.sg.16xlarge scaling group nodes | 0 | Yes | The maximum number of kx.sg.16xlarge Managed kdb scaling group nodes per environment. |
| kx.sg.32xlarge scaling group nodes | 0 | Yes | The maximum number of kx.sg.32xlarge Managed kdb scaling group nodes per environment. |
| kx.sg1.16xlarge scaling group nodes | 0 | Yes | The maximum number of kx.sg1.16xlarge Managed kdb scaling group nodes per environment. |
| kx.sg1.24xlarge scaling group nodes | 0 | Yes | The maximum number of kx.sg1.24xlarge Managed kdb scaling group nodes per environment. |

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Managed kdb volumes | 5 | Yes | The maximum number of Managed kdb volumes per environment. |
| Managed kdb volume read mounts | 5 | Yes | The maximum number of read mounts per Managed kdb volume per environment. |
| Managed kdb volume write mounts | 5 | Yes | The maximum number of write mounts per Managed kdb volume per environment. |
| Managed kdb volume storage | 7730 Gigabytes | Yes | The maximum amount of storage for Managed kdb volumes per environment. |

The following table describes the quotas for kdb dataviews within FinSpace.

| Name | Quota | Adjustable | Description |
|---|---|---|---|
| Managed kdb dataviews | 4500 | Yes | The maximum number of Managed kdb dataviews per environment. |
| Concurrent dataview version processing | 10 | Yes | The maximum number of concurrent Managed kdb dataview version processing. |

# Document history

The following table describes important additions to the Amazon FinSpace documentation.

| Change | Description | Date |
| --- | --- | --- |
| Database maintenance APIs | Added a new database maintenance q API. | Aug 13, 2024 |
| Resource management with scaling groups | Added two new smaller instance types for scaling groups. | Aug 13, 2024 |
| Ingestion APIs | Added two new Ingestion Q APIs. | Jun 17, 2024 |
| Database maintenance | Amazon FinSpace now allows you to perform schema changes to your database. | Mar 19, 2024 |
| Quotas | Added new service quotas for changesets. | Feb 21, 2024 |
| Managed kdb scaling groups | Amazon FinSpace now allows you to create shared compute that you can use to run your clusters on. | Dec 8, 2023 |
| Dataviews for querying data | Amazon FinSpace now allows you to place portions of your Managed kdb Insights object store database onto disk for faster read-only access. | Dec 8, 2023 |
| New Tickerplant clusters | Added new cluster type | Dec 8, 2023 |
| Managed kdb volumes | Added new feature — Managed kdb volumes. | Dec 8, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| New General purpose clusters | Added new cluster type — General purpose (GP) clusters. | Dec 4, 2023 |
| Updated IAM policy | Updated the `AWSServiceRoleForFinSpace` policy to allow PutMetricData calls to AWS/Usage CloudWatch namespace. | Dec 1, 2023 |
| New cache size and types | Introduced new cache size and types for HDB clusters. | October 30, 2023 |
| Allow code updates on running cluster | Added a new feature to allow updating code on running clusters. | October 30, 2023 |
| Region expansion | Amazon FinSpace Managed kdb Insights is available in five new regions. | October 24, 2023 |
| Cluster management q API opeartions | Added new cluster management q API operations in FinSpace q API reference section. | September 25, 2023 |
| New tutorial | Added a tutorial to configure and validate outbound network connectivity through transit gateway. | September 25, 2023 |
| Restructured the user guide navigation | Restructured the user guide navigation to separate Managed Kdb and Dataset browser topics. | September 25, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| New deployment configurations | Added new deployment configurations to the update cluster datases workflow. | August 21, 2023 |
| Network access control lists (ACL) configuration | Updates to create network connection workflow to include Network ACL. | August 21, 2023 |
| Deployment modes | Updates to kdb cluster database worklfow to include deployment modes. | August 21, 2023 |
| Managed kdb Insights | Amazon FinSpace with Managed kdb Insights provides customers with a fully managed service for the latest version of kdb's analytics engine. | June 5, 2023 |
| New IAM policy | To enable access to AWS service and resources, FinSpace uses the service-linked role named `AWSServiceRoleForFinSpace` . | June 5, 2023 |

# AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.