Moody's ANALYTICS

Better Faster Decisions

Moody's DataHub User Guide

DataHub User Guide ver.1

2022.02.24

Contents

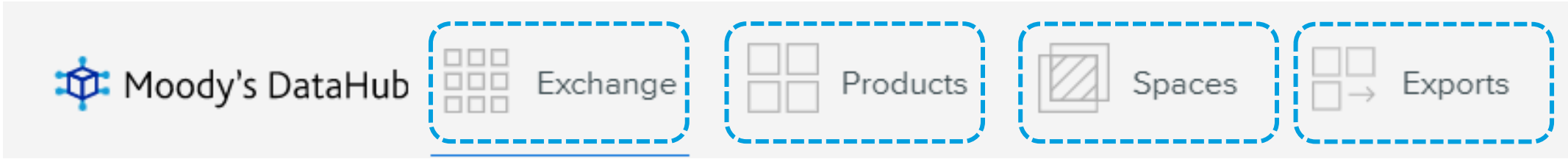
1 Overview of DataHub

DataHub Overview

What is DataHub?

Moody's DataHub is Moody's next generation in data discovery and distribution. DataHub enables informed decision-making by harnessing cloud-based cutting-edge technology to deliver Moody's interconnected datasets. Explore our data, analyze it using industry standard tools, and consume it how you want, when you want. Moody's DataHub provides transparency and control.

DataHub Setup



DataHub is comprised of 4 pages: Exchange, Products, Spaces, and Exports.

Each of these pages have different functionalities that will help to look for, analyze and download the data you need. This User Guide will explain the main functionalities of each of the four pages.

DataHub Functionalities – What can you do on DataHub?

There are various functionalities on the DataHub platform, but the 2 main purposes that clients use DataHub are to ① Download Bulk Data ② Customize and Create your own Datasets.

Downloading Bulk Data

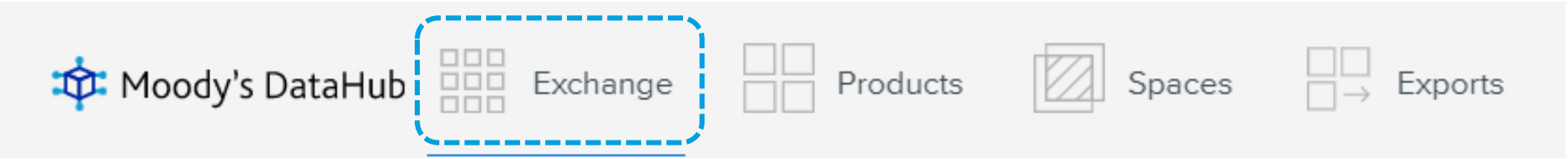
Downloading bulk data means you can export raw data to your endpoint. In order to export the data that you need, please find the appropriate datasets by browsing through Exchange (explanation on pages 7-13) and upon finding the data set that you would like to export, export the datasets by following the instructions under Exports (pages 14-25).

Creating your own datasets

In addition to the preexisting Moody's datasets, you are also able to export datasets you have customized and created for your own analytics purposes. Follow the instructions under Spaces (pages 26-40) to learn how to create a new datasets and the instructions under Products (pages 41-50) to publish your newly created dataset. After your datasets has been created, you can export it under Exports (pages 14-25).

2 Exchange

Exchange

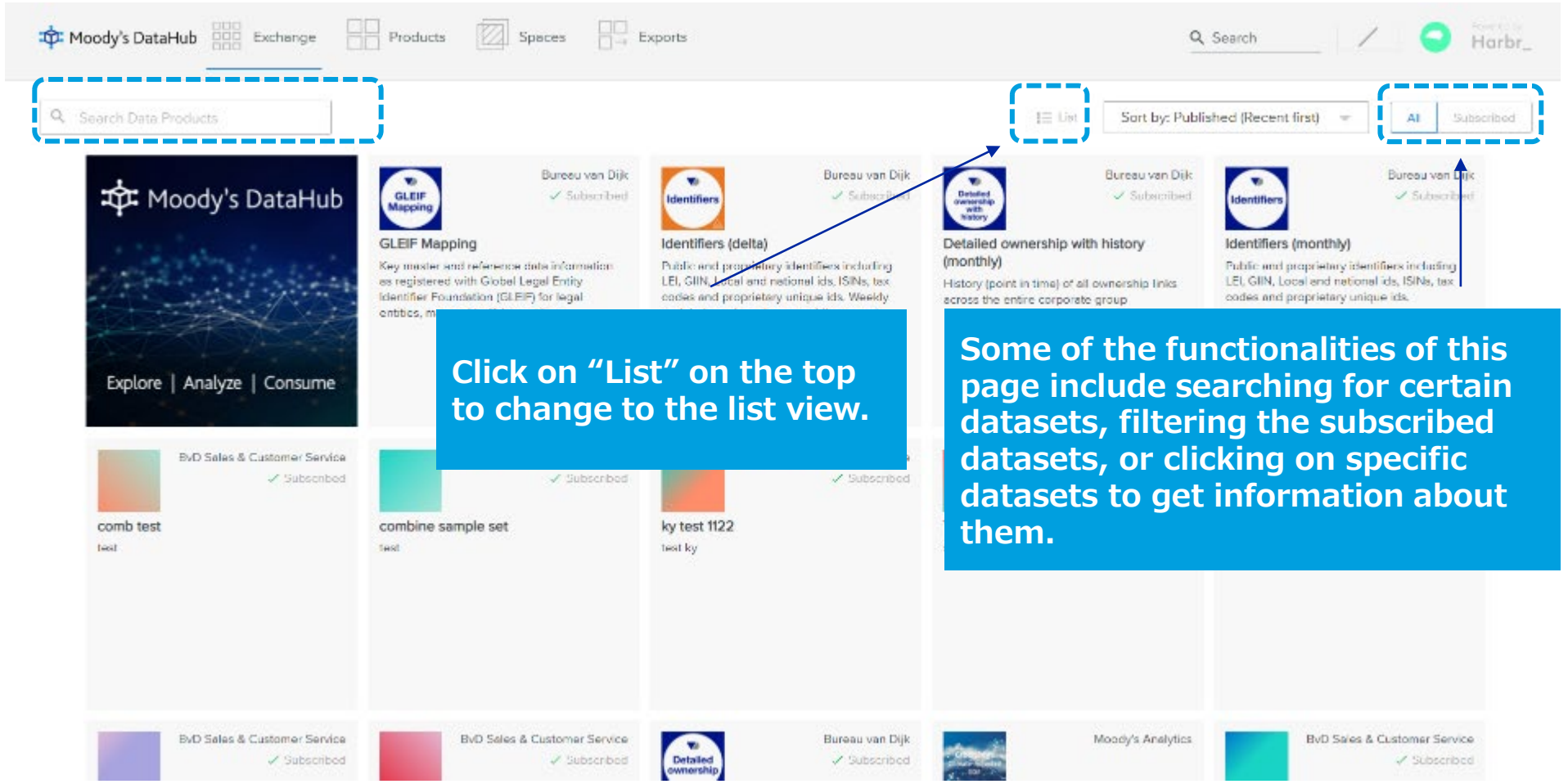


Exchange on DataHub

The first important page on DataHub is the Exchange page. The Exchange is a centralized location for the Moody's datasets available on Moody's DataHub. It allows you to explore all of Moody's data on a centralized platform, including datasets you may not have subscribed to. Embedded data previews and definition give you the information you need to leverage the data in your own analyses. The exchange also lets you see any engineered datasets you have published using the platform.

Exchange – Main Page

Under the "Exchange" tab, you will be able to see all the datasets.



Click on "List" on the top to change to the list view.

Some of the functionalities of this page include searching for certain datasets, filtering the subscribed datasets, or clicking on specific datasets to get information about them.

Exchange – List View

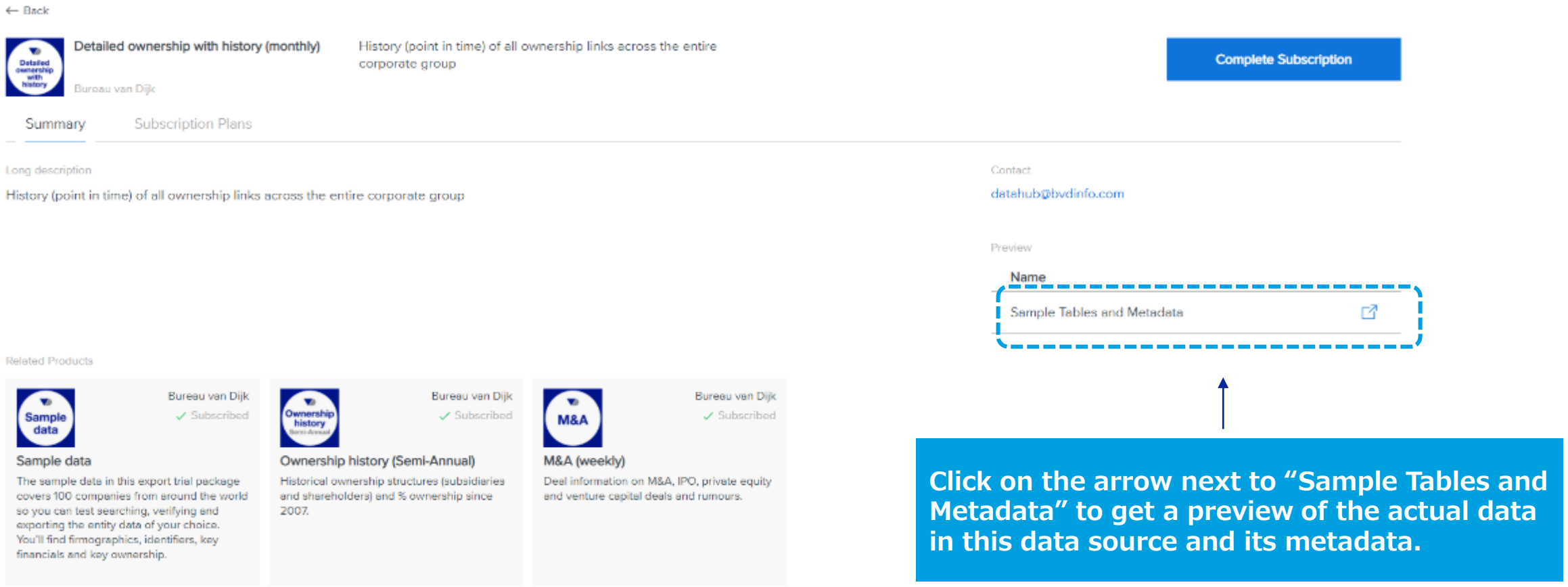
The List view allows you to see the datasets as a list. Use the view that is easier for you to find and browse through the datasets.

Click on "Grid" to change back to the Grid view.

Name	Organization	Subscribed	Description
GLEIF Mapping	Bureau van Dijk	✓	Key master and reference data information as registered with Global Legal Entity Identifier Foundation (GLEIF) for legal entities, mapped to Orbis entities.
Identifiers (delta)	Bureau van Dijk	✓	Public and proprietary identifiers including LEI, GIIN, Local and national ids, ISINs, tax codes and proprietary unique ids. Weekly update based on changes of the record.
Detailed ownership with history (monthly)	Bureau van Dijk	✓	History (point in time) of all ownership links across the entire corporate group
Identifiers (monthly)	Bureau van Dijk	✓	Public and proprietary identifiers including LEI, GIIN, Local and national ids, ISINs, tax codes and proprietary unique ids.
comb test	BvD Sales & Customer Service	✓	test
combine sample set	BvD Sales & Customer Service	✓	test
ky test 1122	BvD Sales & Customer Service	✓	test ky
Tile	BvD Sales & Customer Service	✓	short
Orbis sample 20211102	BvD Sales & Customer Service	✓	A few date point in Orbis datasets
Export address	BvD Sales & Customer Service	✓	Only for export

Exchange – Data Product Page

As an example, this is what it looks when you click on a certain dataset. This page contains useful information such as the description of the data source, whom to contact, related datasets and a preview of the data.

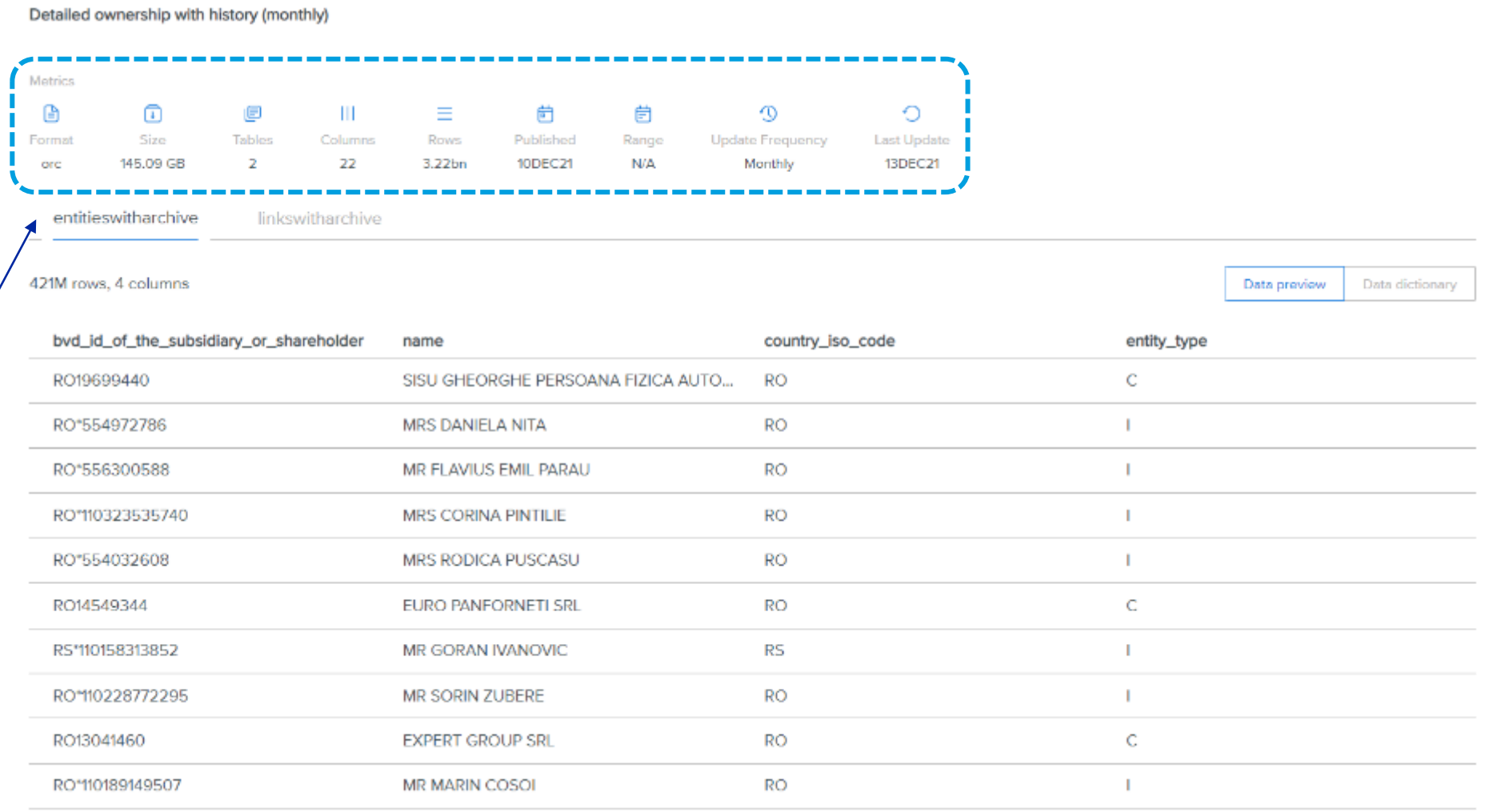


Click on the arrow next to "Sample Tables and Metadata" to get a preview of the actual data in this data source and its metadata.

Exchange – Sample Tables and Metadata

In the "Sample Tables and Metadata" page, you will be able to see a preview of a part of the table in the data source and its metadata.

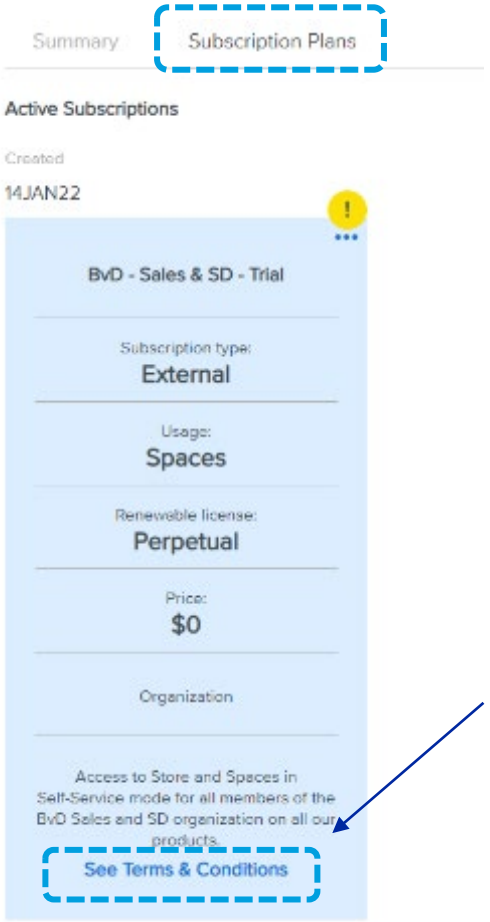
This page also gives you critical information such as the size, number of tables, columns and rows, date published, update frequency, and when the last update took place.



bvd_id_of_the_subsidiary_or_shareholder	name	country_iso_code	entity_type
RO19699440	SISU GHEORGHE PERSOANA FIZICA AUTO...	RO	C
RO*554972786	MRS DANIELA NITA	RO	I
RO*556300588	MR FLAVIUS EMIL PARAU	RO	I
RO*110323535740	MRS CORINA PINTILIE	RO	I
RO*554032608	MRS RODICA PUSCASU	RO	I
RO14549344	EURO PANFORNETI SRL	RO	C
RS*110158313852	MR GORAN IVANOVIC	RS	I
RO*110228772295	MR SORIN ZUBERE	RO	I
RO13041460	EXPERT GROUP SRL	RO	C
RO*110189149507	MR MARIN COSOI	RO	I

Exchange – Subscription Plans

Please note that you will not be able to export a dataset unless you have completed the subscription.



Please click on “Terms & Conditions” and agree to our terms.

Once you have accepted our terms and conditions and completed your subscription to the dataset, you will see the completion sign on the top right-hand side of the page.

3 Exports

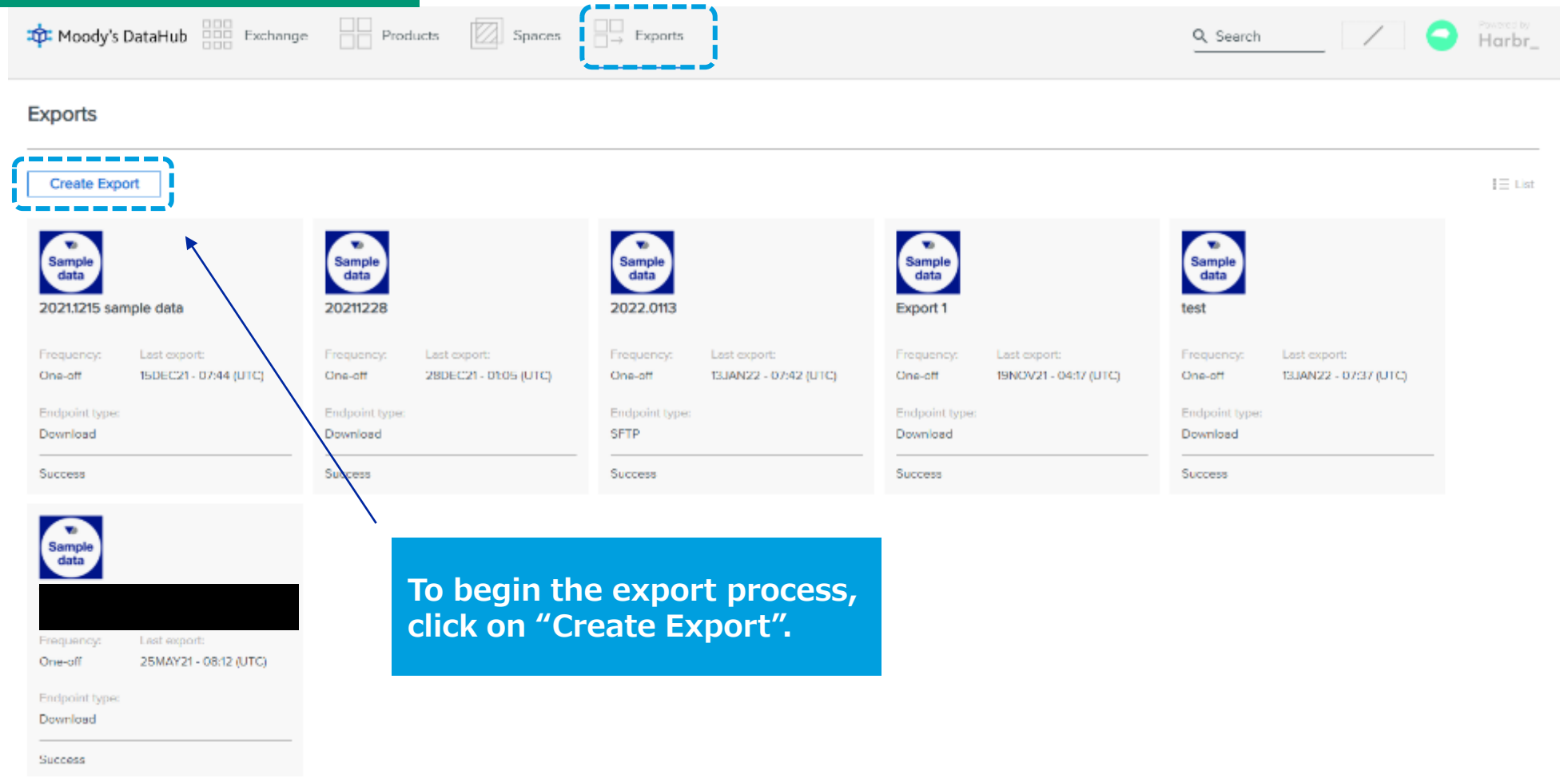
Exports

Exports on DataHub

The next important page on DataHub is Exports. The Exports page allows you to export data from the platform for internal system and process integration. The platform supports the multiple export methods (Download, SFTP, S3 and GCS) and file formats (CSV, Avro, Parquet, ORC).

Exports – Main Page

Under the "Exports" tab, you will be able to see the history of your past exports.

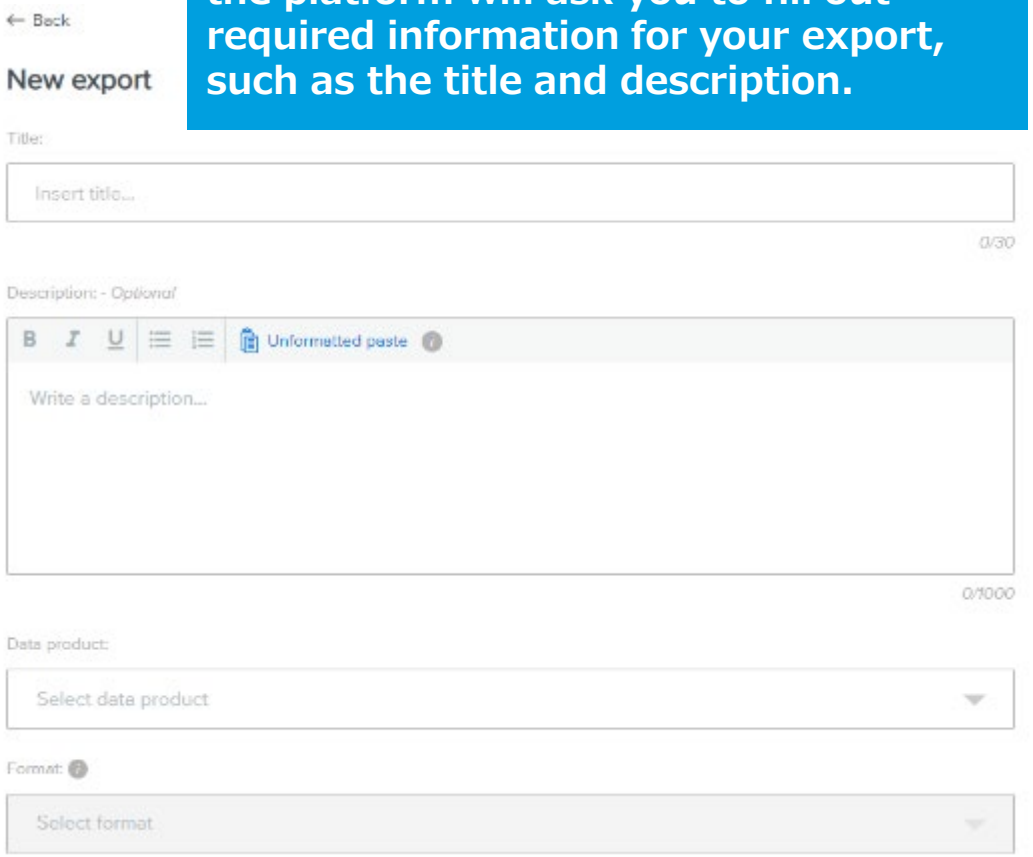


To begin the export process, click on "Create Export".

Exports – Create Export

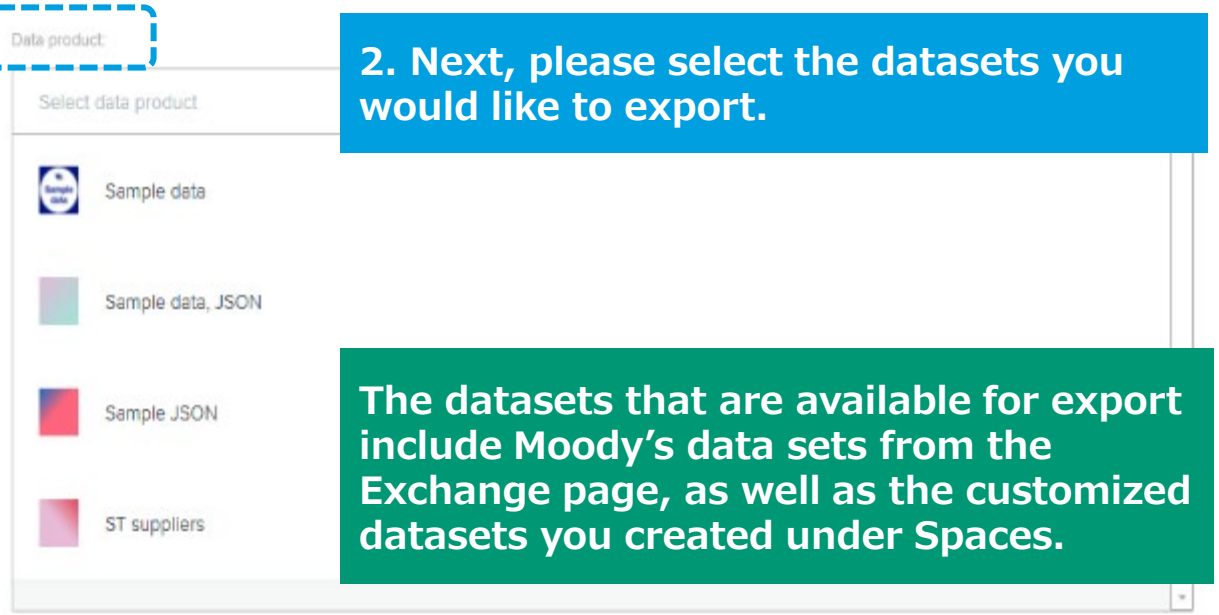
1.

1. After clicking on create new export, the platform will ask you to fill out required information for your export, such as the title and description.



2.

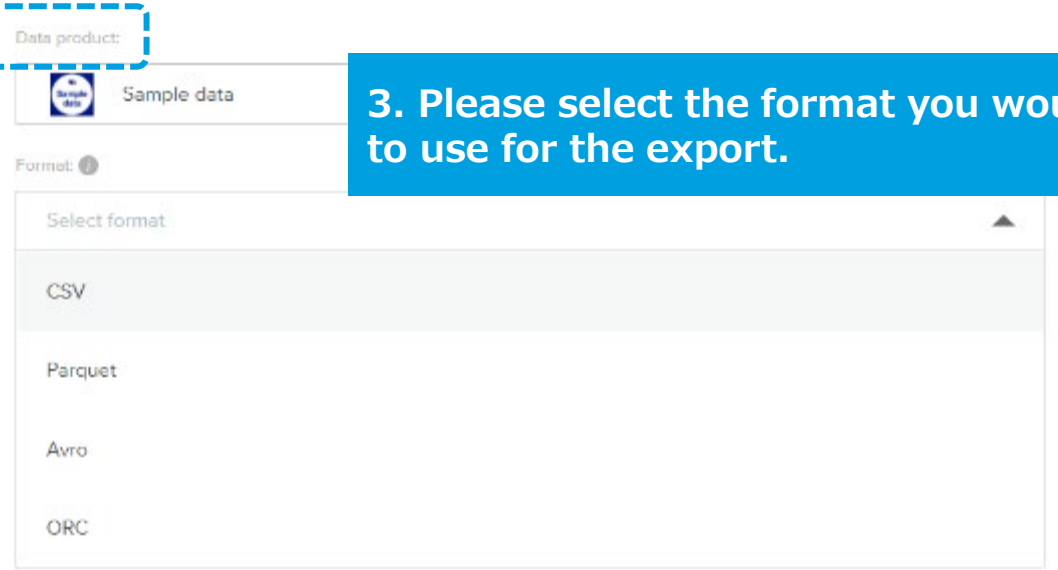
2. Next, please select the datasets you would like to export.



The datasets that are available for export include Moody's data sets from the Exchange page, as well as the customized datasets you created under Spaces.

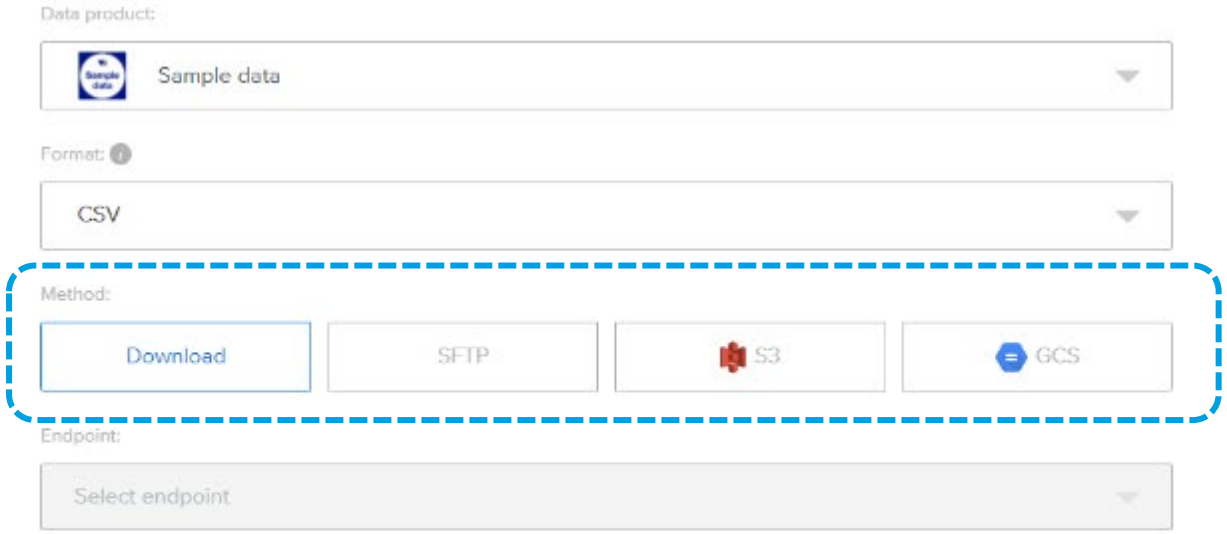
3.

3. Please select the format you would like to use for the export.



Exports – Create Export (cont.)

4.



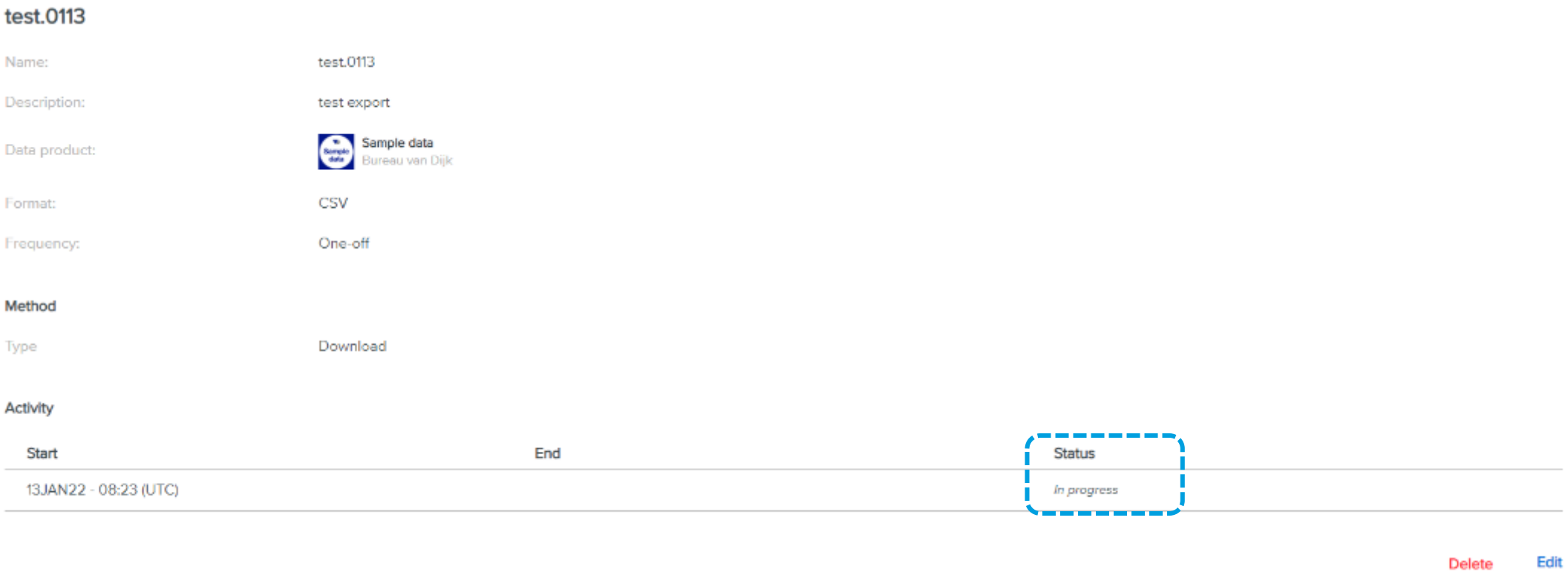
4. Lastly, please select the method of export. By selecting "Download", the selected datasets will be downloaded to your device. By selecting "SFTP/S3/GCS", the datasets would be exported to the selected cloud environment.

5. Click "Create" to finalize your export options.

5.

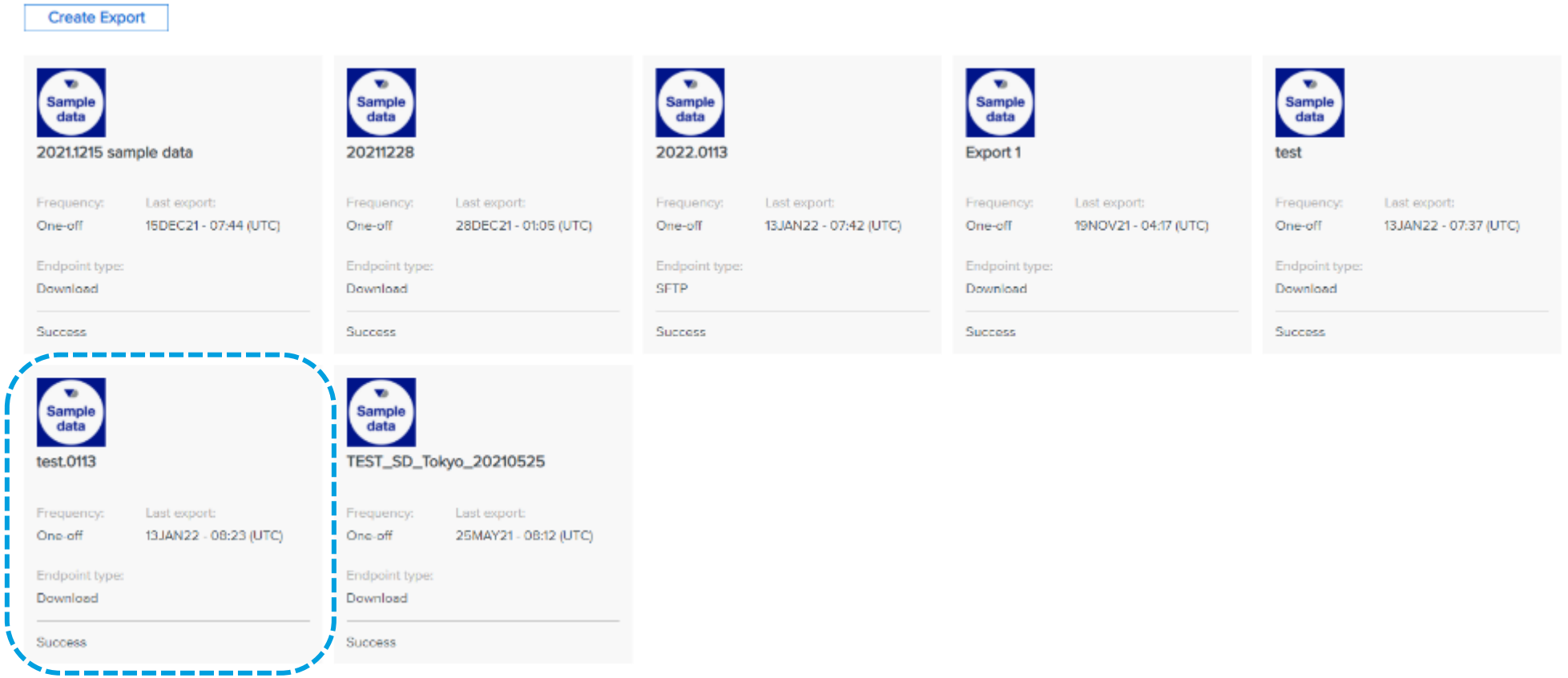
Exports – Create Export (cont.)

After clicking on “Create”, the platform will show you the summary of your export. Note that the export status is “in progress”. To complete this step, please refresh the page or go back to the Exports tab.

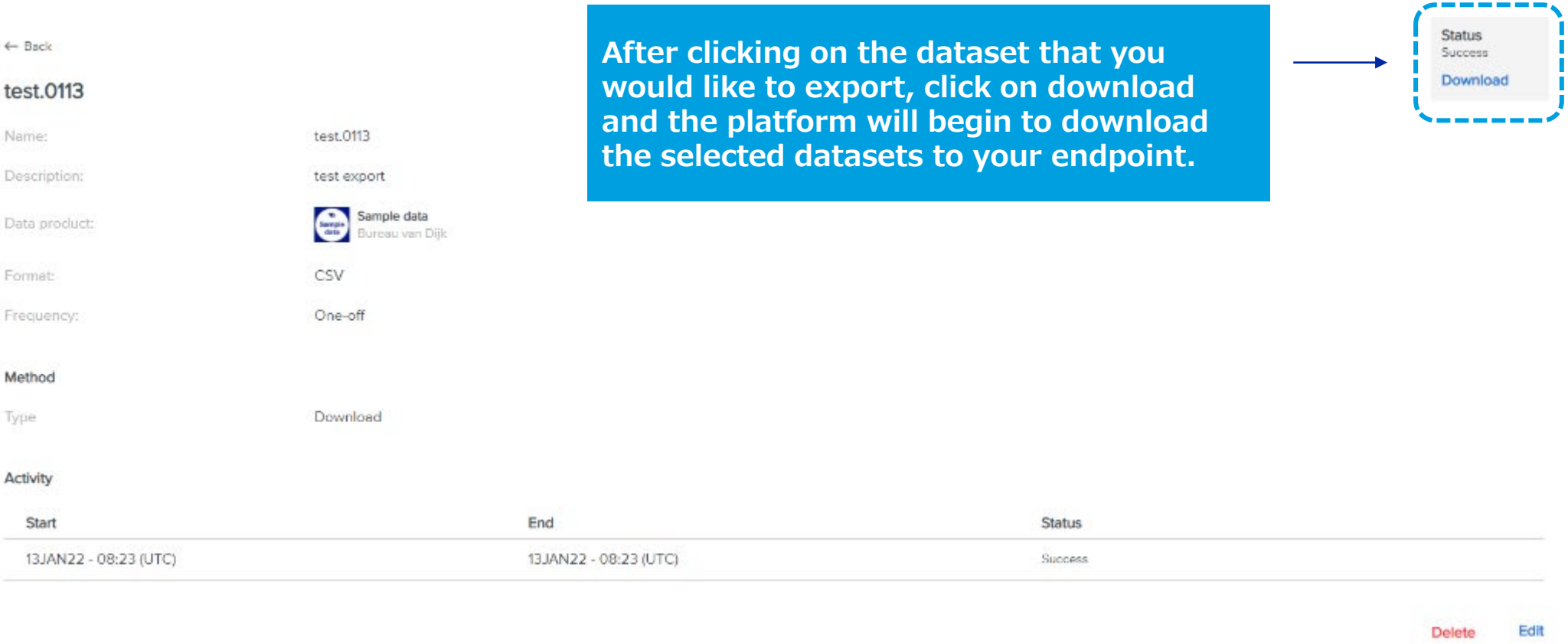


Exports – Create Export (cont.)

After the export has been created, it will now be shown under the Export tab. Please note that the dataset has not yet been downloaded. Click on the grid you would like to export.

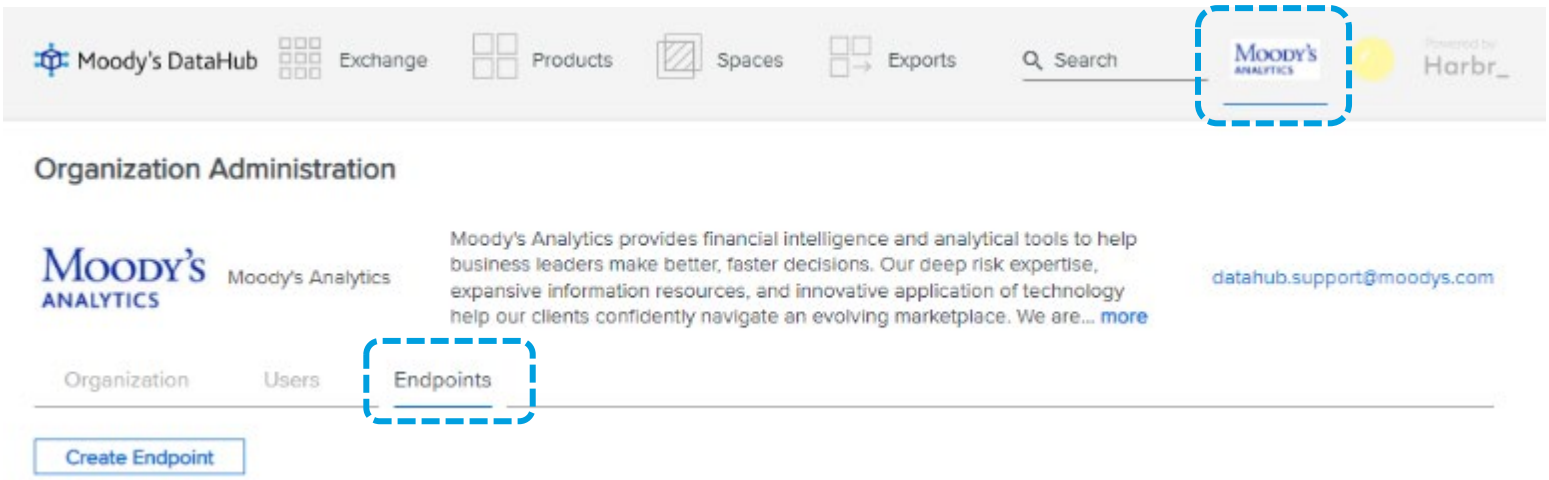


Exports – Create Export (cont.)



After clicking on the dataset that you would like to export, click on download and the platform will begin to download the selected datasets to your endpoint.

Exports – Endpoints

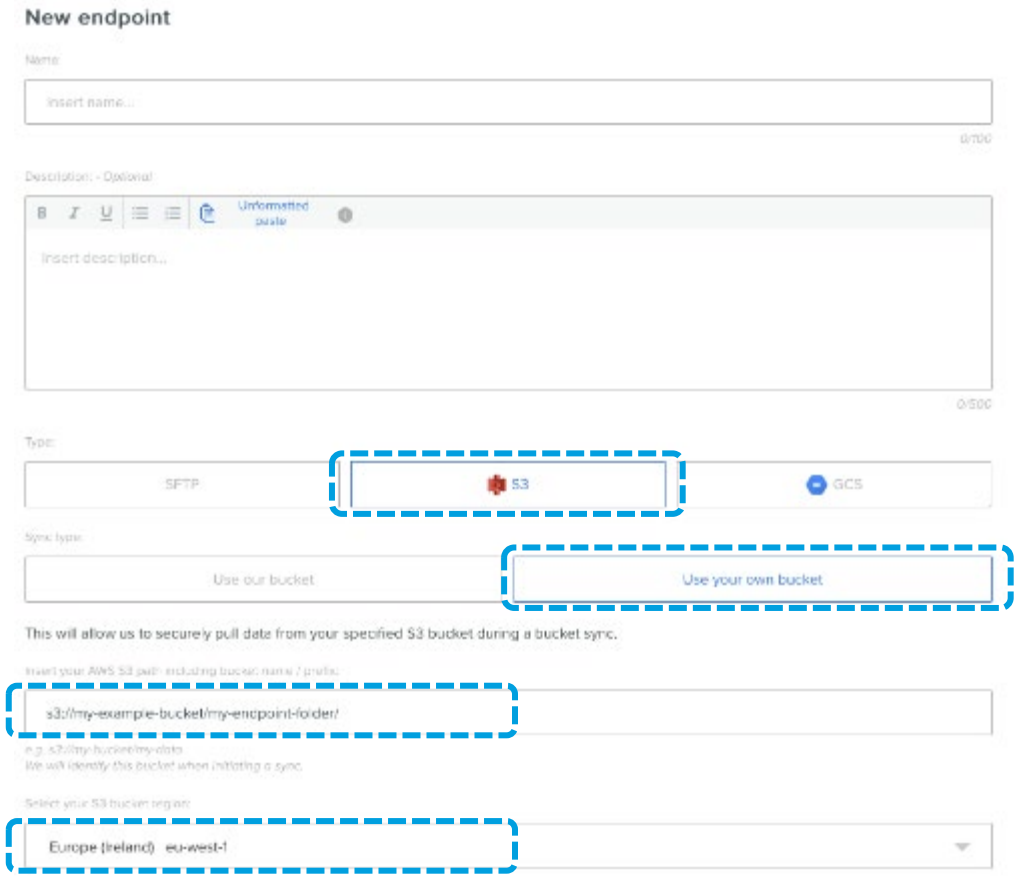


Endpoints on DataHub

Endpoints are used by the Export service to deliver Data Product content to external target locations. The platform supports the following endpoints: Amazon S3, Google Cloud Platform Bucket Storage, SFTP (Export Only).

Exports – Create Amazon S3 Endpoints

1. **1. Provide the S3 location (bucket name and region) within your AWS account that you wish to associate with the endpoint**

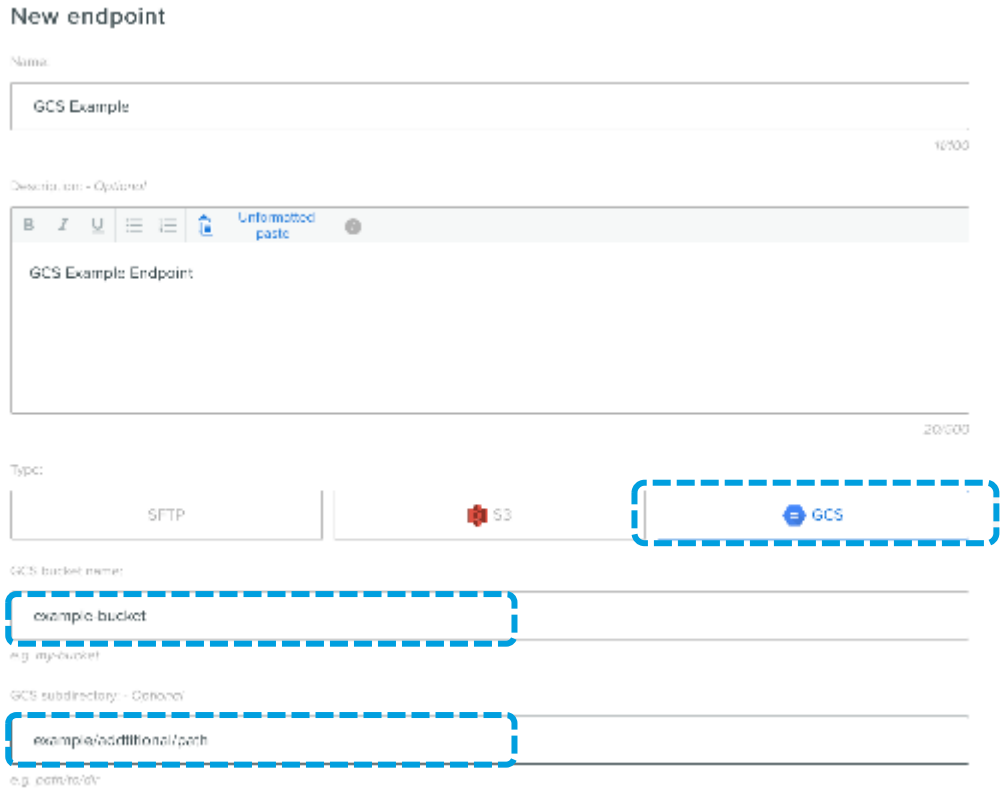


2. **2. Copy the generated S3 bucket policy into your AWS account**

```
{
 "Version": "2012-10-17",
 "Id": "rztGVYbCROeJiT27EnJaYQ",
 "Statement": [
  {
   "Sid": "rztGVYbCROeJiT27EnJaYQ_1",
   "Effect": "Allow",
   "Principal": {"AWS": "arn:aws:iam::111111111111:root"},
   "Action": [
    "s3:GetBucketLocation",
    "s3:ListBucket"
   ],
   "Resource": [
    "arn:aws:s3:::my-example-bucket"
   ]
  },
  {
   "Sid": "rztGVYbCROeJiT27EnJaYQ_2",
   "Effect": "Allow",
   "Principal": {"AWS": "arn:aws:iam::111111111111:root"},
   "Action": [
    "s3:GetObject",
    "s3:PutObject",
    "s3:PutObjectAcl",
    "s3:DeleteObject"
   ],
   "Resource": "arn:aws:s3:::my-example-bucket/my-endpoint-folder/*"
  },
  {
   "Sid": "rztGVYbCROeJiT27EnJaYQ_3",
   "Effect":"Deny",
   "Principal": {"AWS": "arn:aws:iam::111111111111:root"},
   "Action":"s3:PutObject",
   "Resource": "arn:aws:s3:::my-example-bucket/my-endpoint-folder/*",
   "Condition": {
    "StringNotEquals": {"s3:x-amz-acl":"bucket-owner-full-control"}
   }
  }
 ]
}
```

Exports – Create GCS Storage Bucket Endpoints

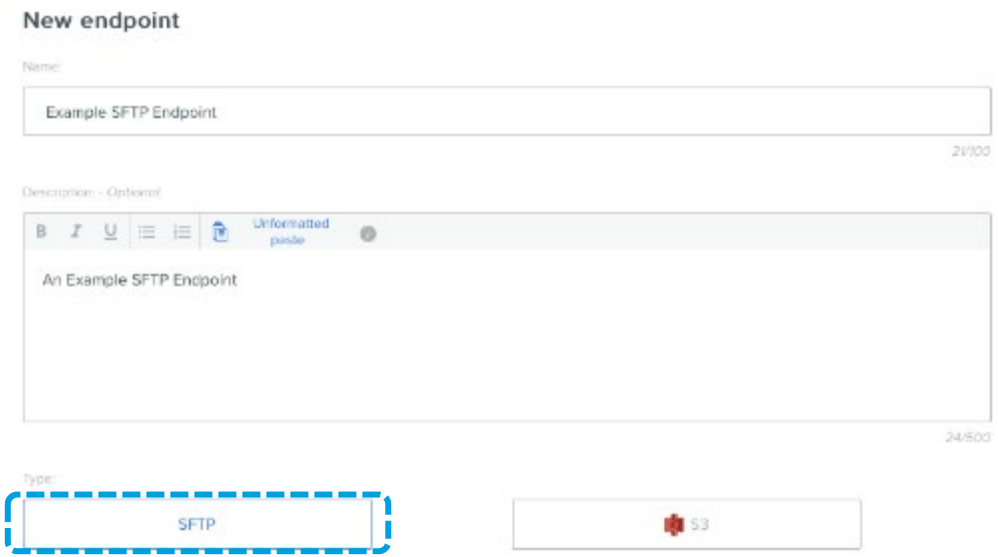
1. **1. Provide the GCS Storage Bucket location (bucket name and subdirectory) within GCP projects that you wish to associate with the endpoint**



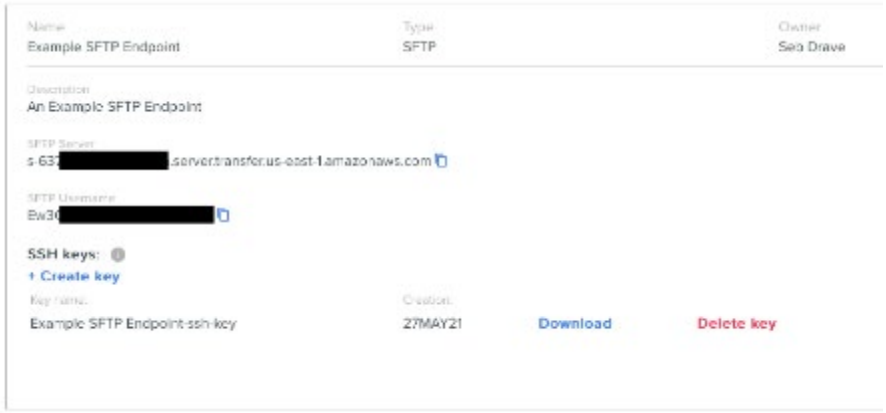
2. **2. Permission the service account onto the bucket specified in your endpoint definition with the highlighted roles**

Exports – Create SFTP Endpoints

1. **1. Provide the name and description of the SFTP endpoint. Select type as 'SFTP '**

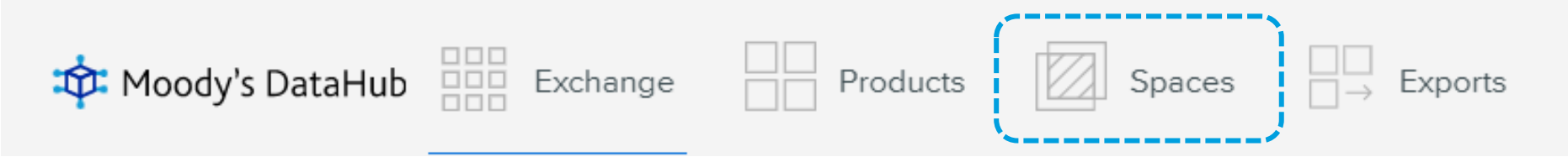


2. **2. The platform will generate and communicate the SFTP access credentials**



4 Spaces

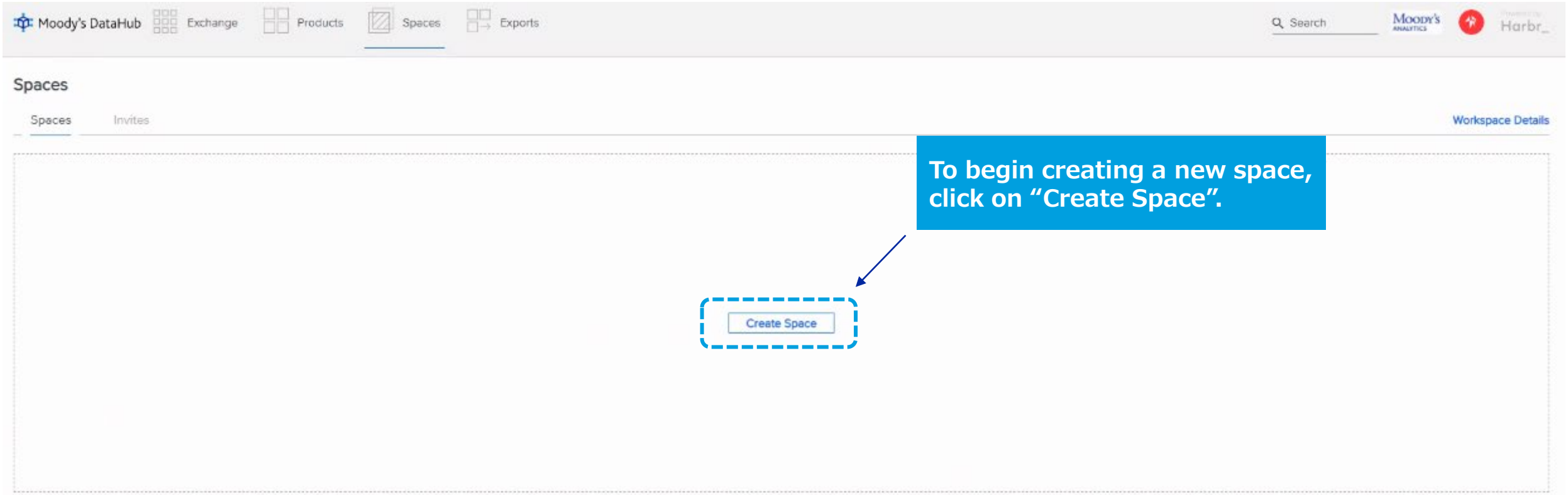
Spaces



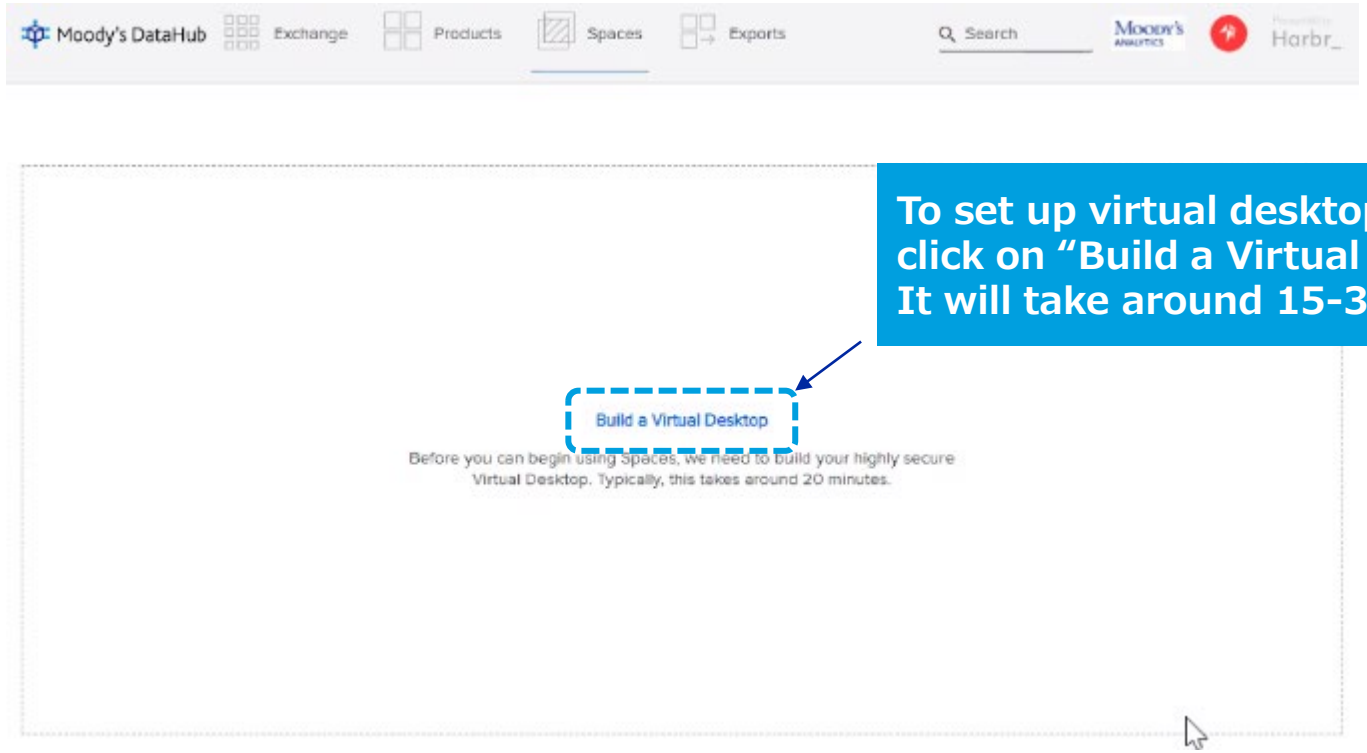
Spaces on DataHub

Another page on DataHub is Spaces. Spaces are secure collaborative environments that easily bring people, tools, data and models together, enabling secure opportunities to work across content boundaries and create engineered datasets and services. Spaces provide an optimal full stack solution for the effective execution of data projects. Using a combination of industry leading data security, configuration, deployment, and access management, Spaces provides flexibility in a controlled framework.

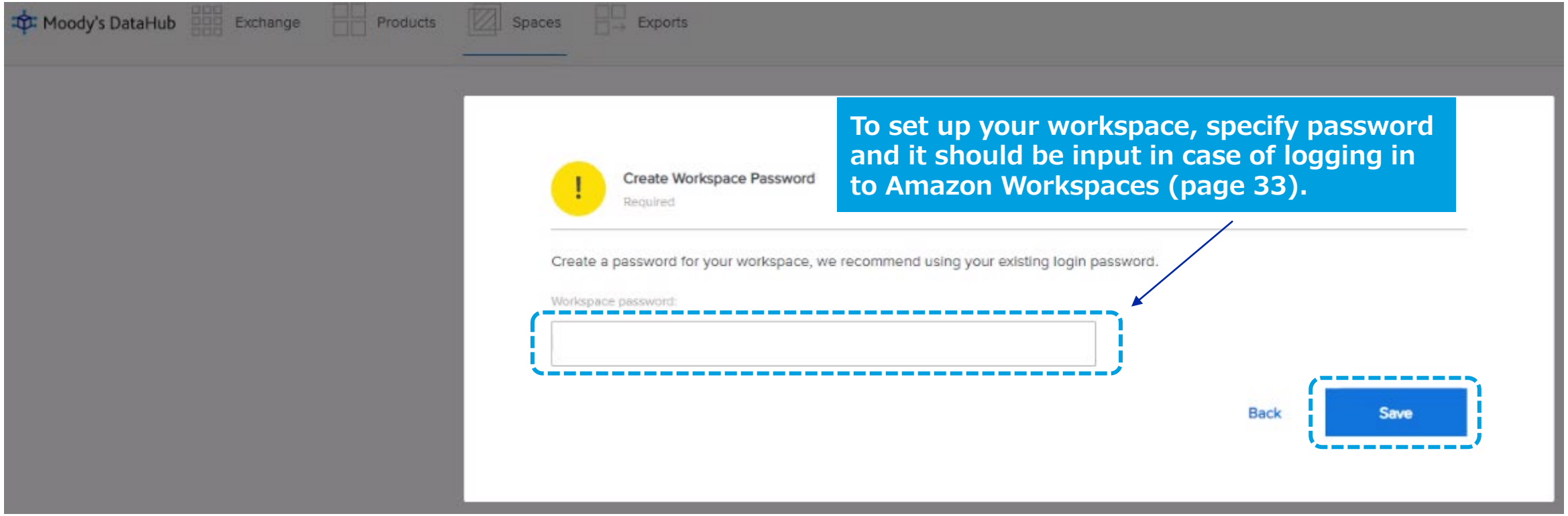
<For the first time> Spaces – Create Space



<For the first time> Spaces – Create Space



<For the first time> Spaces – Create Space

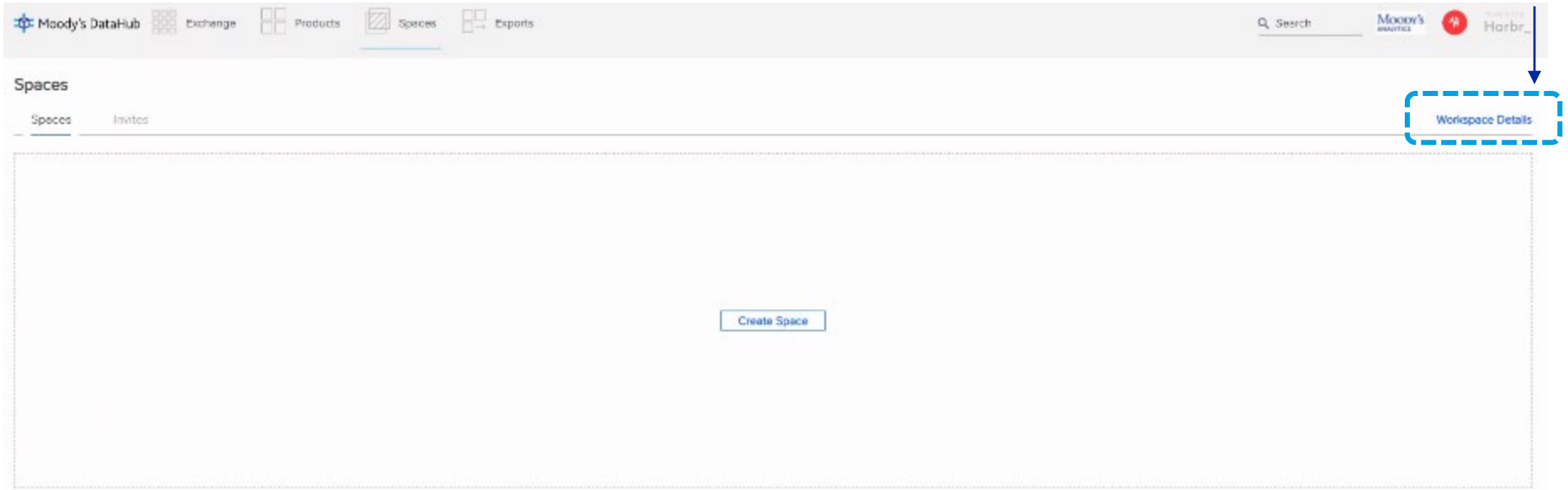


Spaces – Amazon WorkSpaces

Please note that although you have created the space on the Moody's DataHub platform, you will not be able to work on the space on DataHub. You can work on your space only on Amazon WorkSpaces .

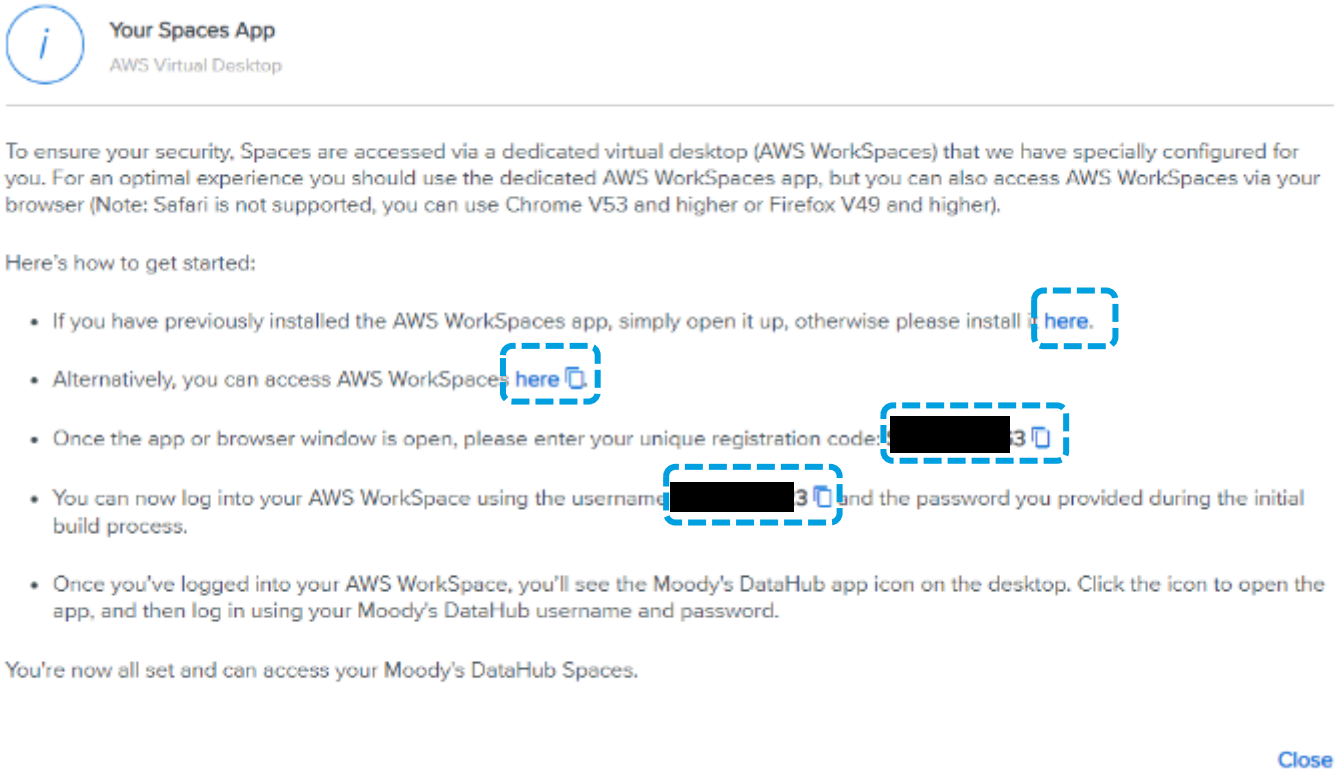
After creating a space on the DataHub platform, please access your Amazon WorkSpaces to work on the space you have created. On Amazon WorkSpaces, you will be able to customize or merge datasets by coding and programming with various analytics tools.

The "Workspace Details" page contains information on installing Amazon WorkSpaces and accessing the Moody's DataHub Spaces. Click on "Workspace Details" for further information.



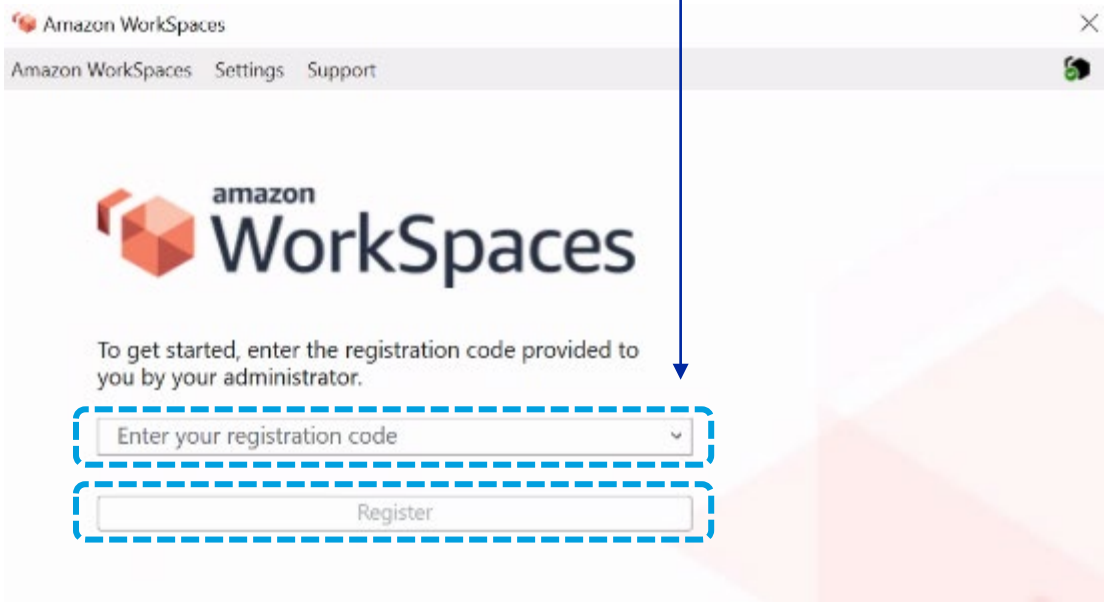
Spaces – WorkSpaces Details

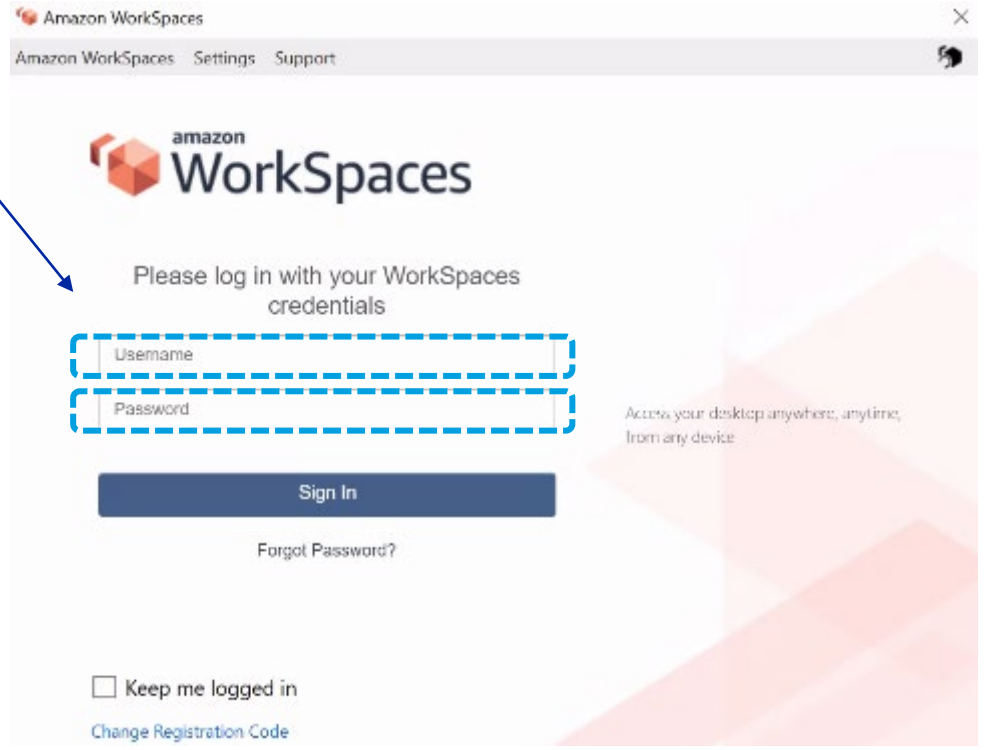
Upon clicking on “Workspace Details”, the instructions on how to install Amazon WorkSpaces and the relevant ID information will be shown.



Spaces – WorkSpaces Details

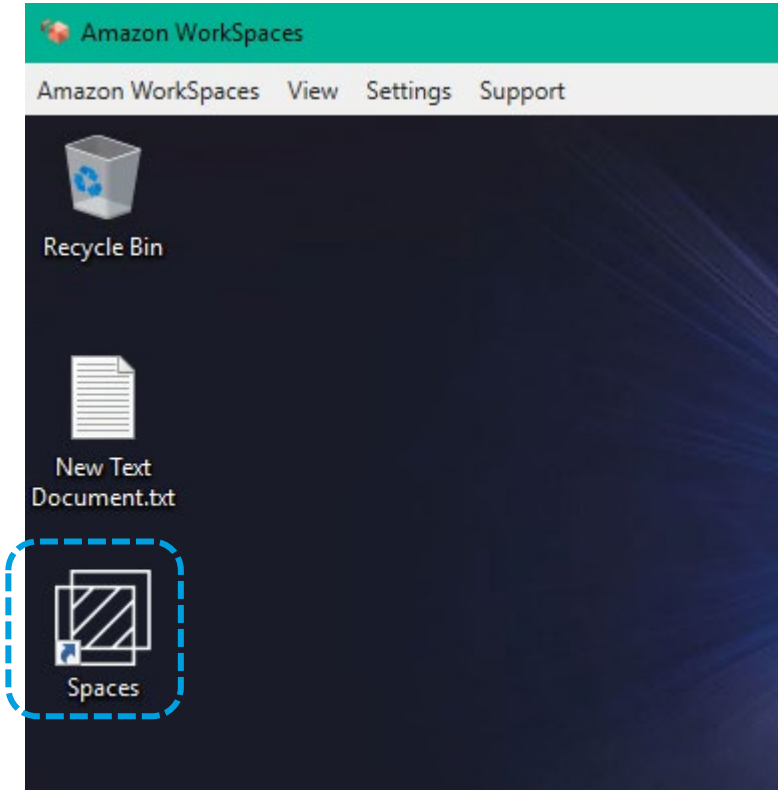
After installing Amazon Workspaces on your machine, key the relevant information in previous page into "registration code", "Username" and "Password".

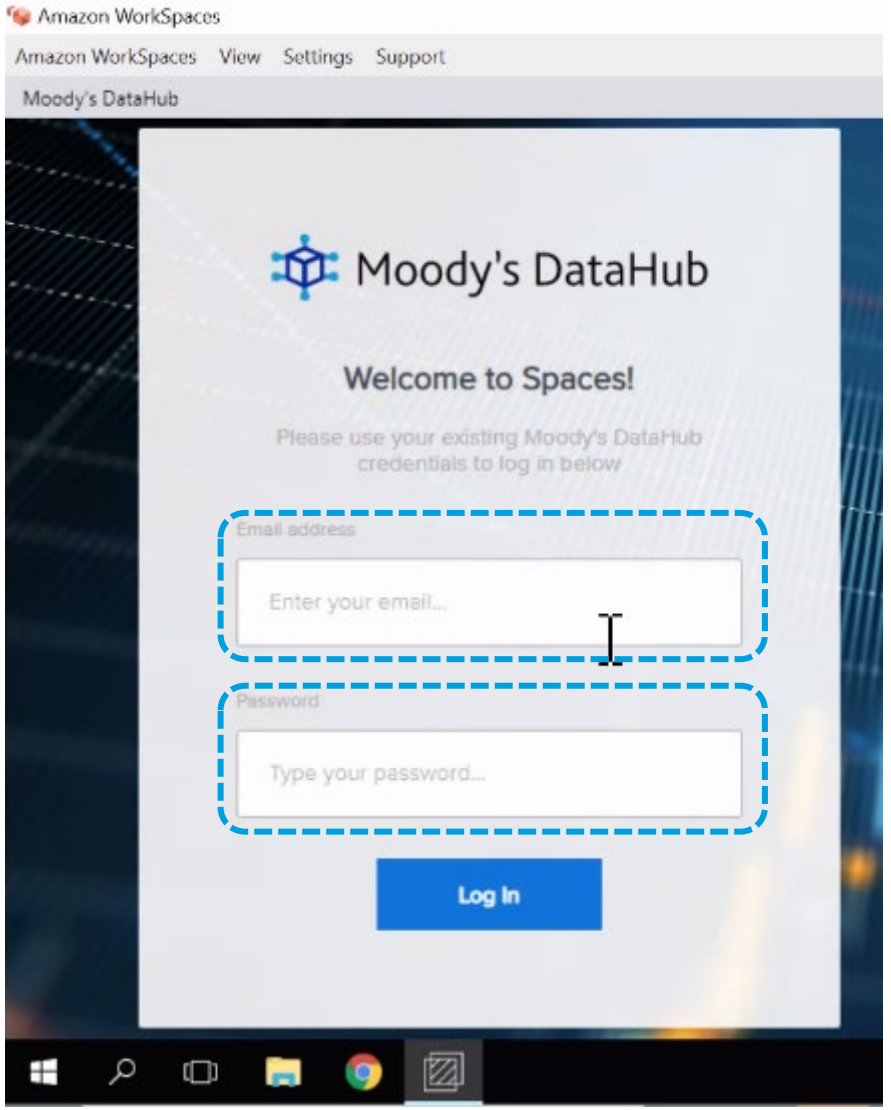




Spaces – on Amazon WorkSpaces

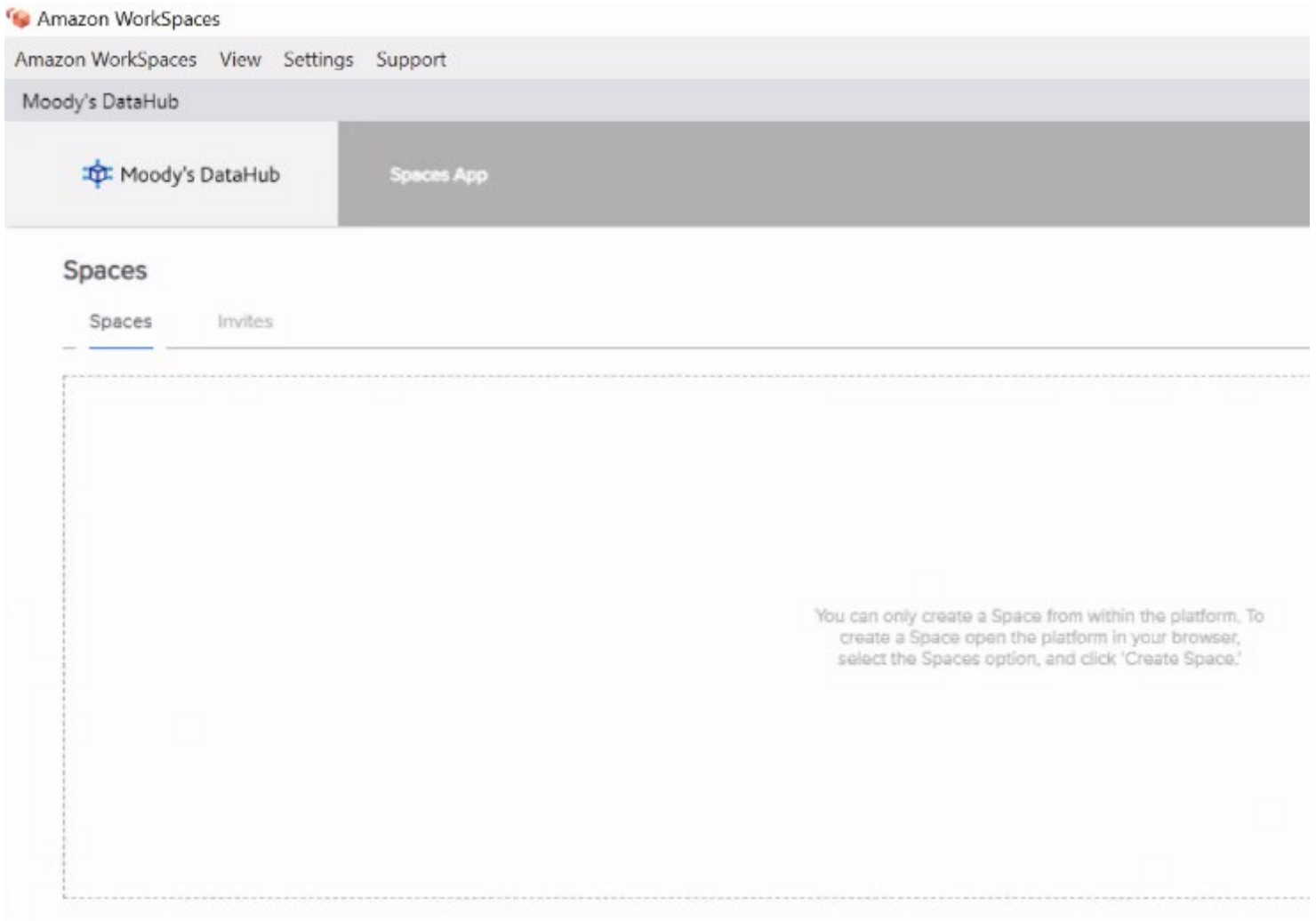
**After launching Windows OS on Amazon Workspaces,
Click “Spaces” icon on the desktop.
And enter your credentials in the DataHub login page.**





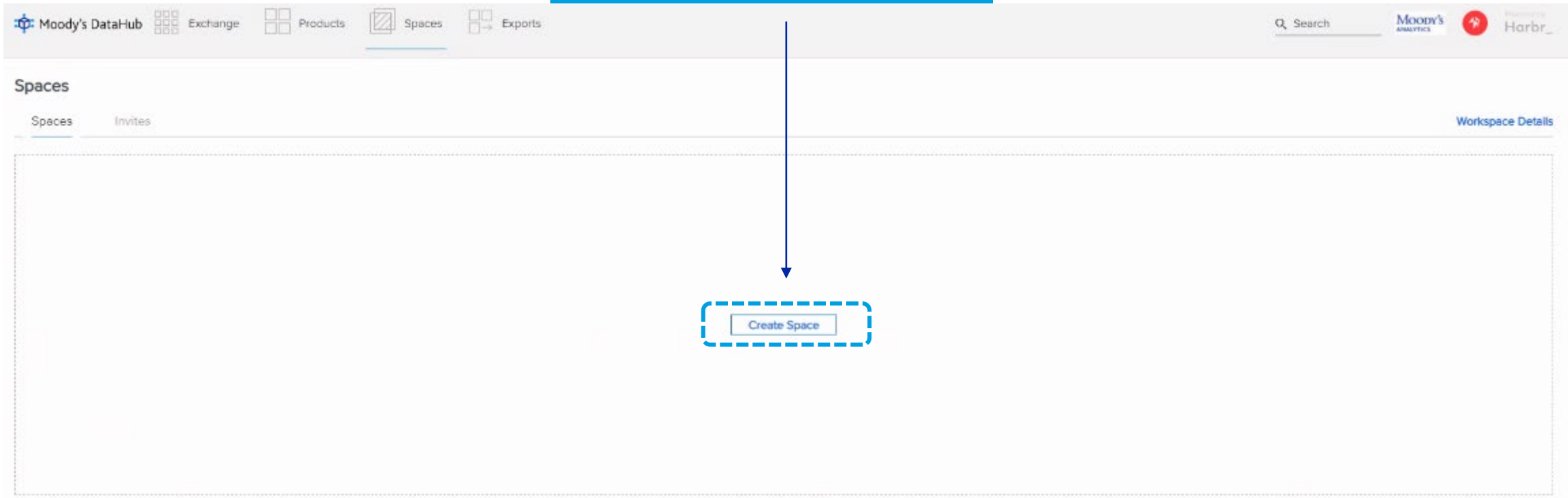
Spaces – on Amazon WorkSpaces

“Spaces” will be launched in Amazon WorkSpaces.



Spaces – Create Space

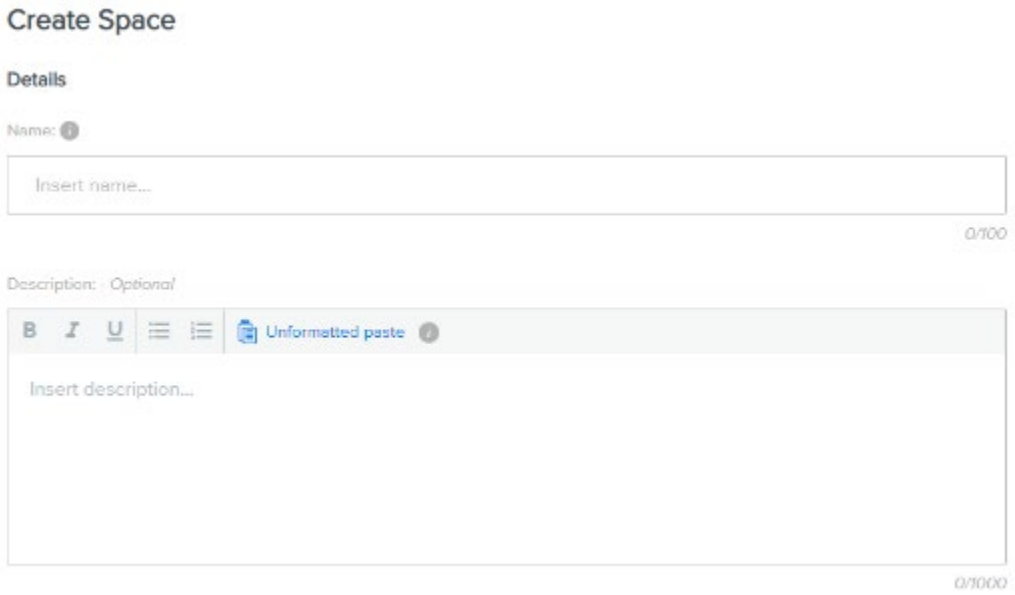
To begin creating a new space, click on “Create Space”.



Spaces – Create Space

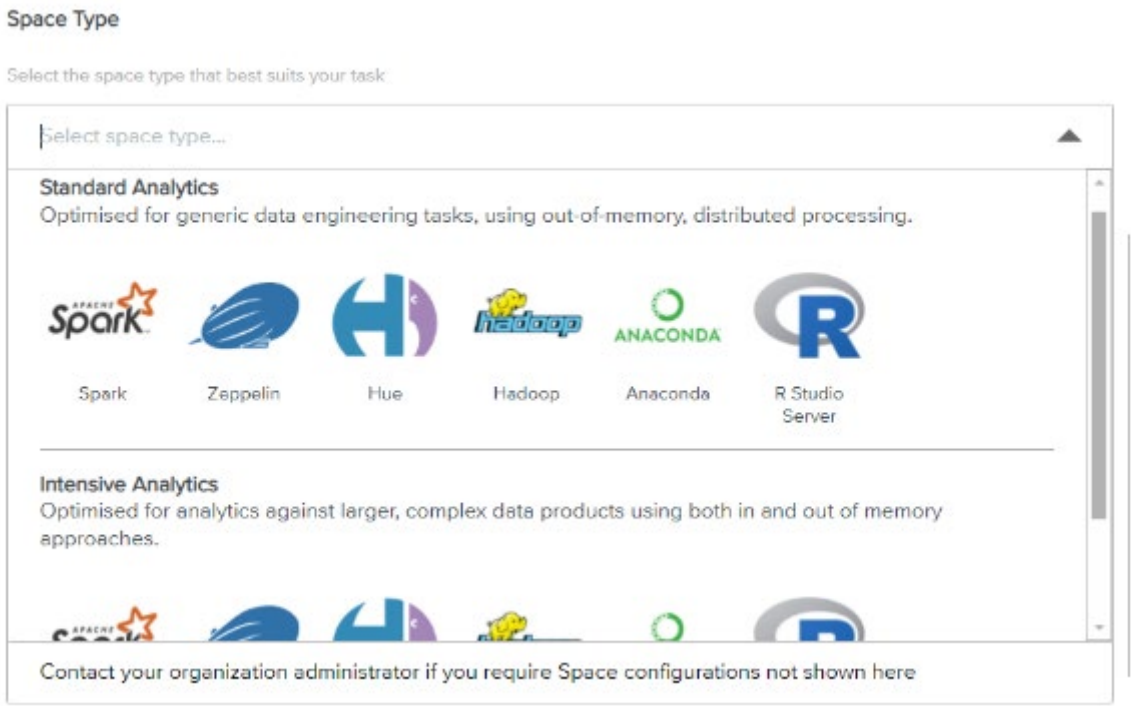
1.

1. The platform will ask you to fill out required information to create a new space. Begin by creating a title and writing a description for your space.



2.

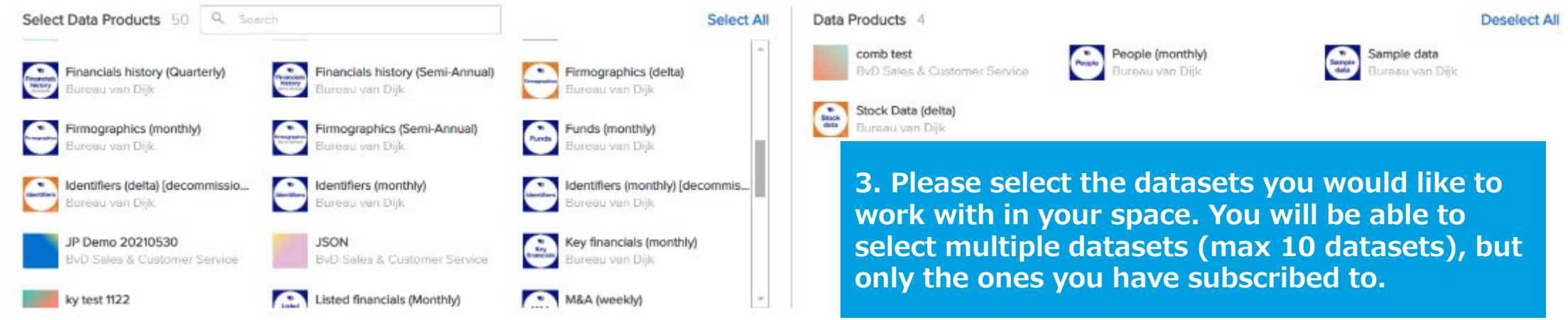
2. Please select "Standard Analytics" or "Intensive Analytics" for your space type. Please select "Intensive Analytics" only for working with larger and complex datasets. You should normally select the Standard version.



Spaces – Create Space (cont.)

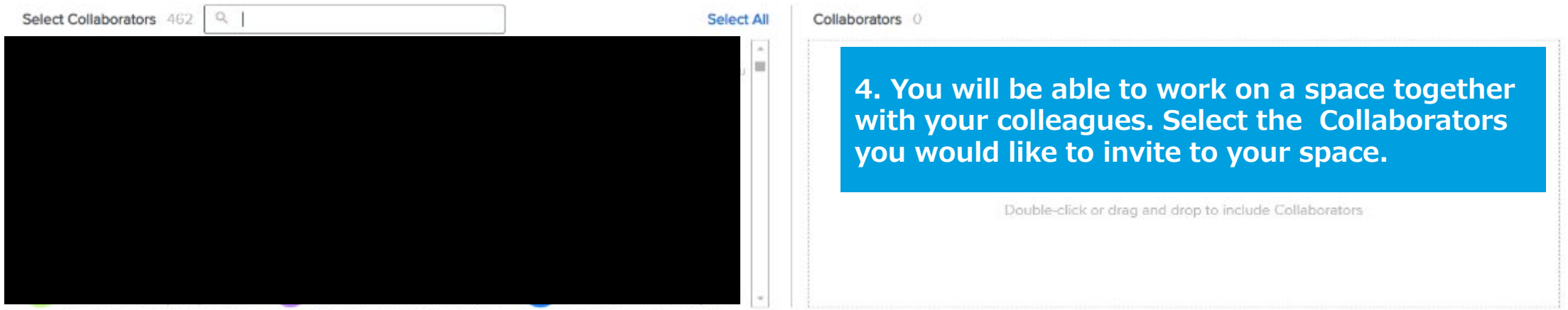
You will be able to select Data Products and Collaborators by double-clicking or dragging and dropping the grid to the box on the right-hand side.

3.



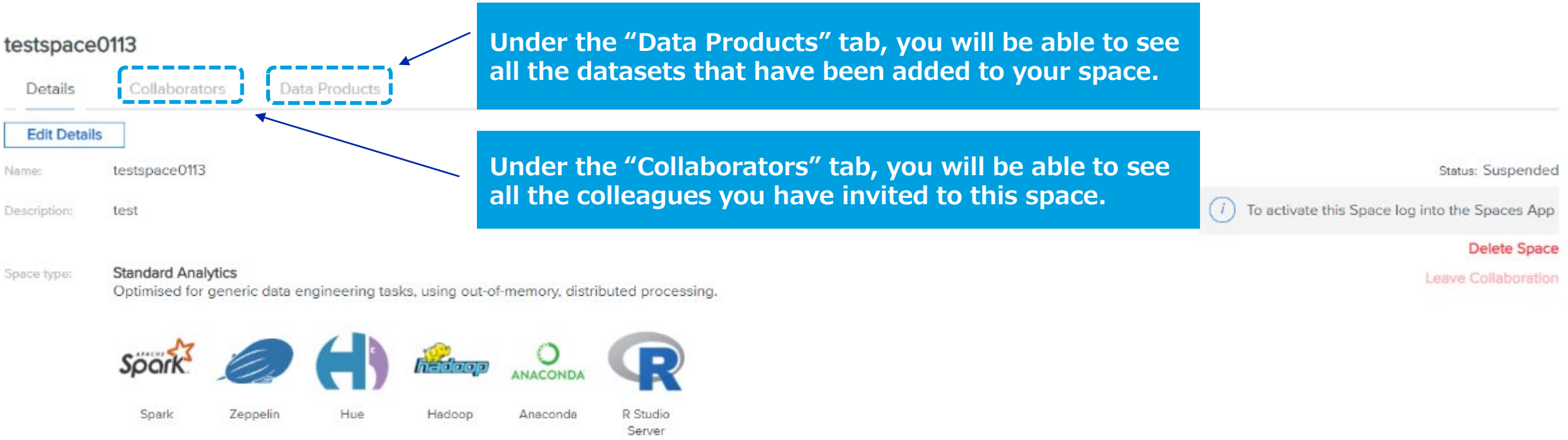
3. Please select the datasets you would like to work with in your space. You will be able to select multiple datasets (max 10 datasets), but only the ones you have subscribed to.

4.



4. You will be able to work on a space together with your colleagues. Select the Collaborators you would like to invite to your space.

Spaces – New Space



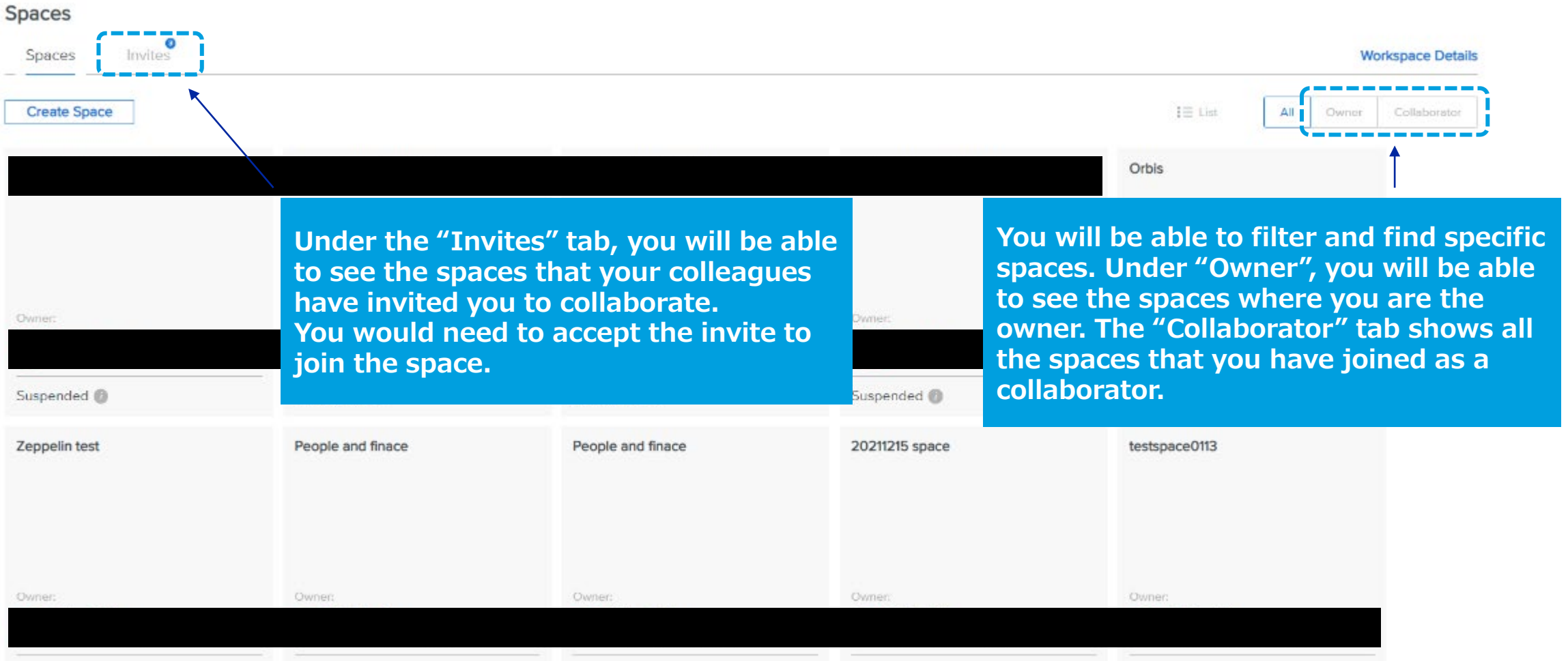
Under the "Data Products" tab, you will be able to see all the datasets that have been added to your space.

Under the "Collaborators" tab, you will be able to see all the colleagues you have invited to this space.

After you have created the space, your newly created space will show under the Spaces tab. Click on the space and you will be able to see all the information that you have entered.
Please refer to Sample Queries (pages 51-63) when you want to take an action to leverage "Spaces" feature.

Spaces – Main Page

Under the "Spaces" tab, you will be able to see all the spaces that you have created.



**Under the "Invites" tab, you will be able to see the spaces that your colleagues have invited you to collaborate.
You would need to accept the invite to join the space.**

You will be able to filter and find specific spaces. Under "Owner", you will be able to see the spaces where you are the owner. The "Collaborator" tab shows all the spaces that you have joined as a collaborator.

5 Products

Products

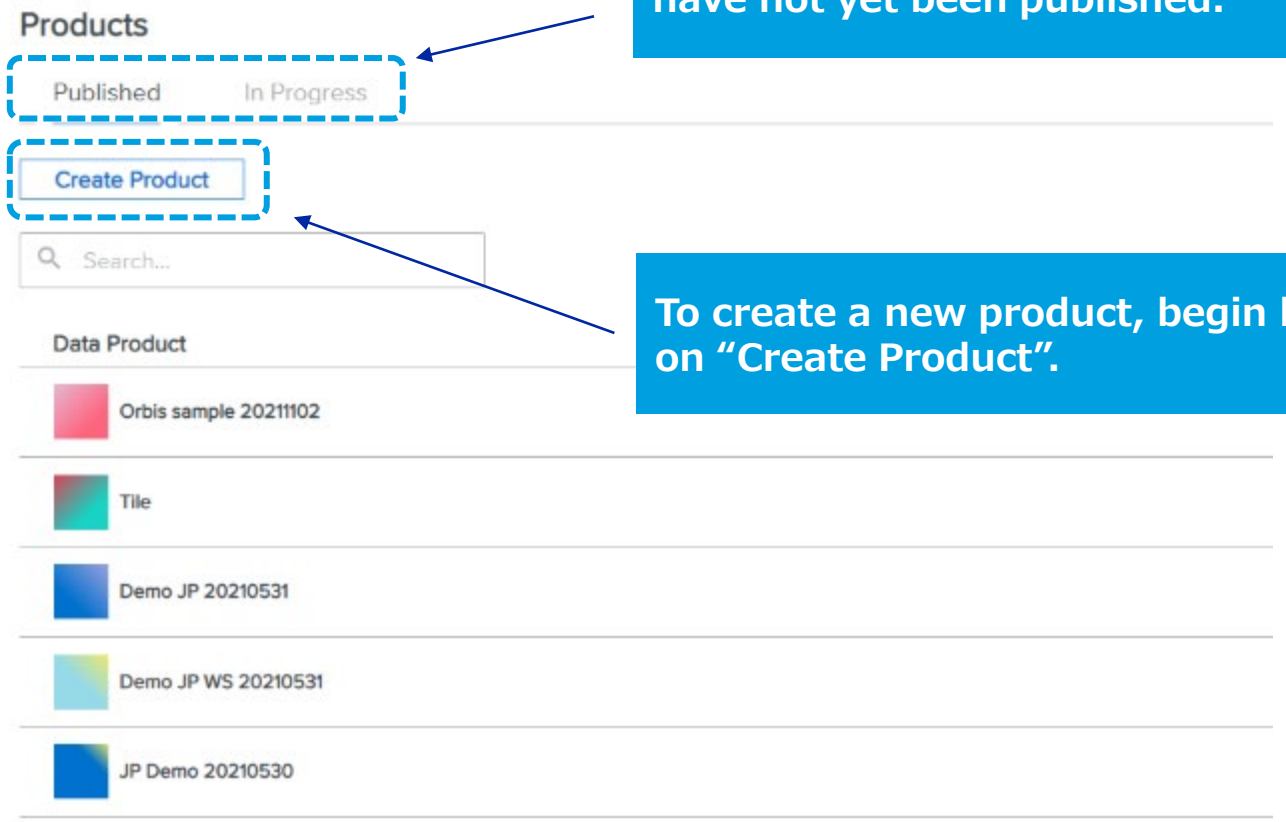
Products on DataHub

The Products page shows all the datasets that you have published. The Products refer to the datasets that you have created and customized in Spaces. By publishing the Products, you will be able to export it to your endpoint, as well as to use in Spaces as a dataset.

Products – Main Page

Under the “Products” tab, you will be able to see the datasets that you have created.

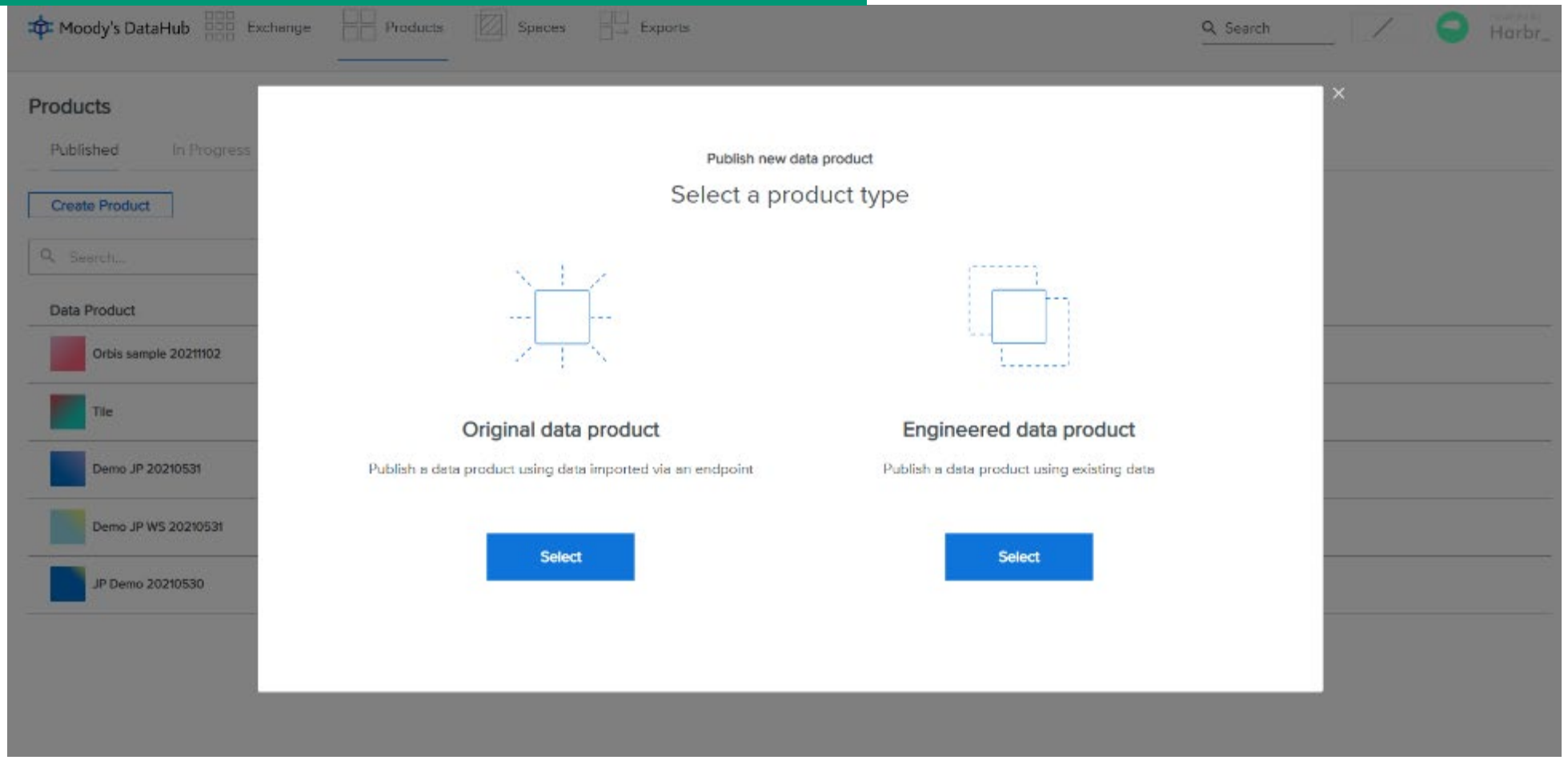
Under the “Published” tab, you can see the datasets that you have already published. Under the “In Progress” tab, you can see the datasets that you have created but have not yet been published.



To create a new product, begin by clicking on “Create Product”.

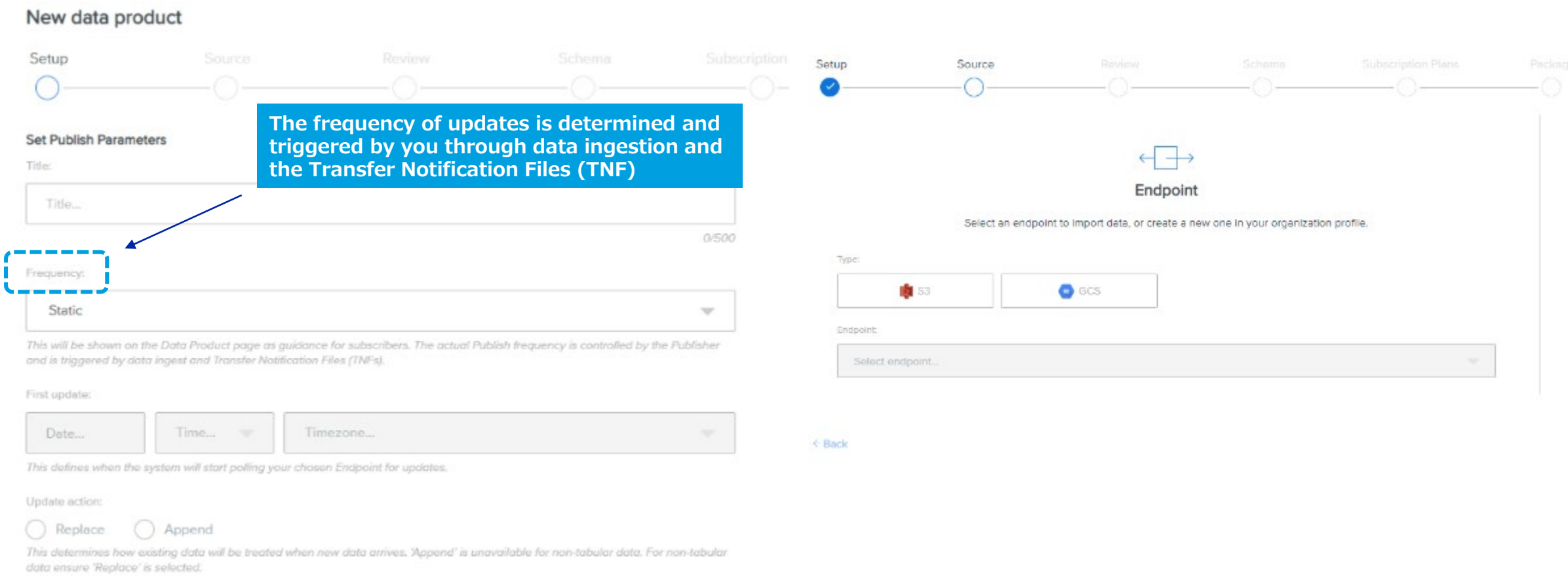
Products – Create Product

As a first step, read the descriptions and select if you would like to create an “Original data product” or an “Engineered data product”.

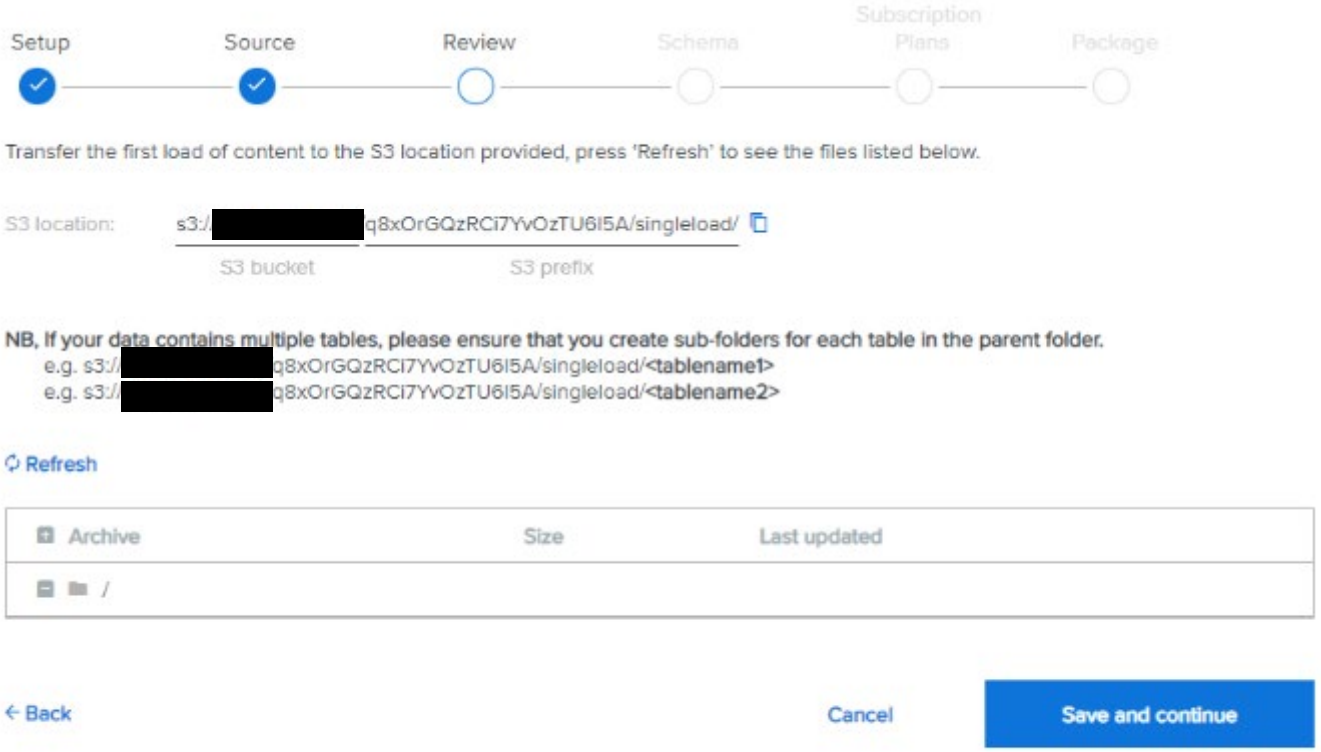


Products – Create “Original data product”

“Original data product” is used to publish your own dataset (outside of Datahub) onto the DataHub platform. It ingests data from your own cloud storage (S3 and GCS).



Products – Create “Original data product” (cont.)



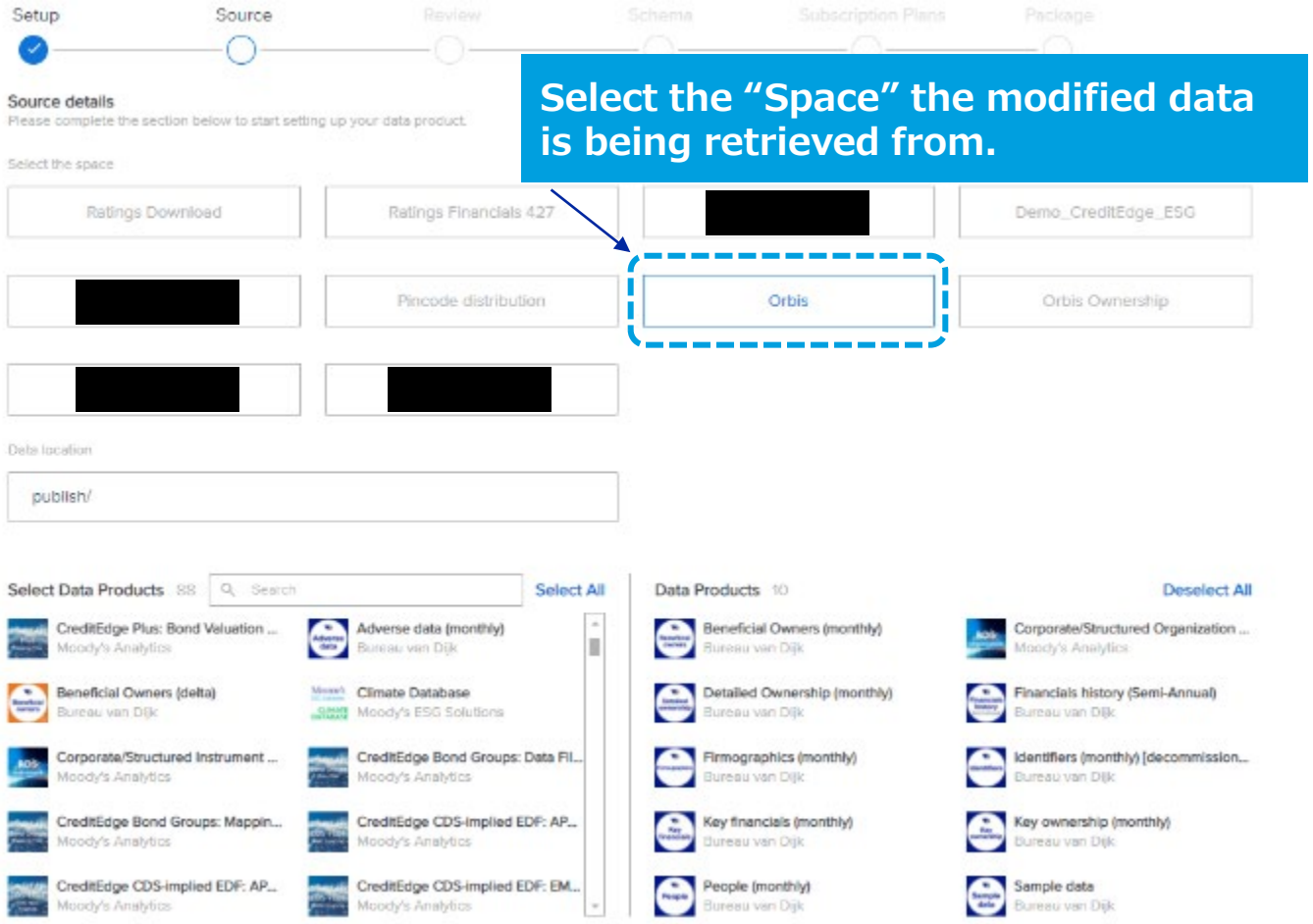
Follow the instructions on how to connect your source and transfer your data to the DataHub platform.

Once the data discovery process is completed, the original data product will be displayed on the “Exchange” page.

As per other datasets, you may choose to “Export” or “Create Space” with this new original dataset.

Products – Create “Engineered data product”

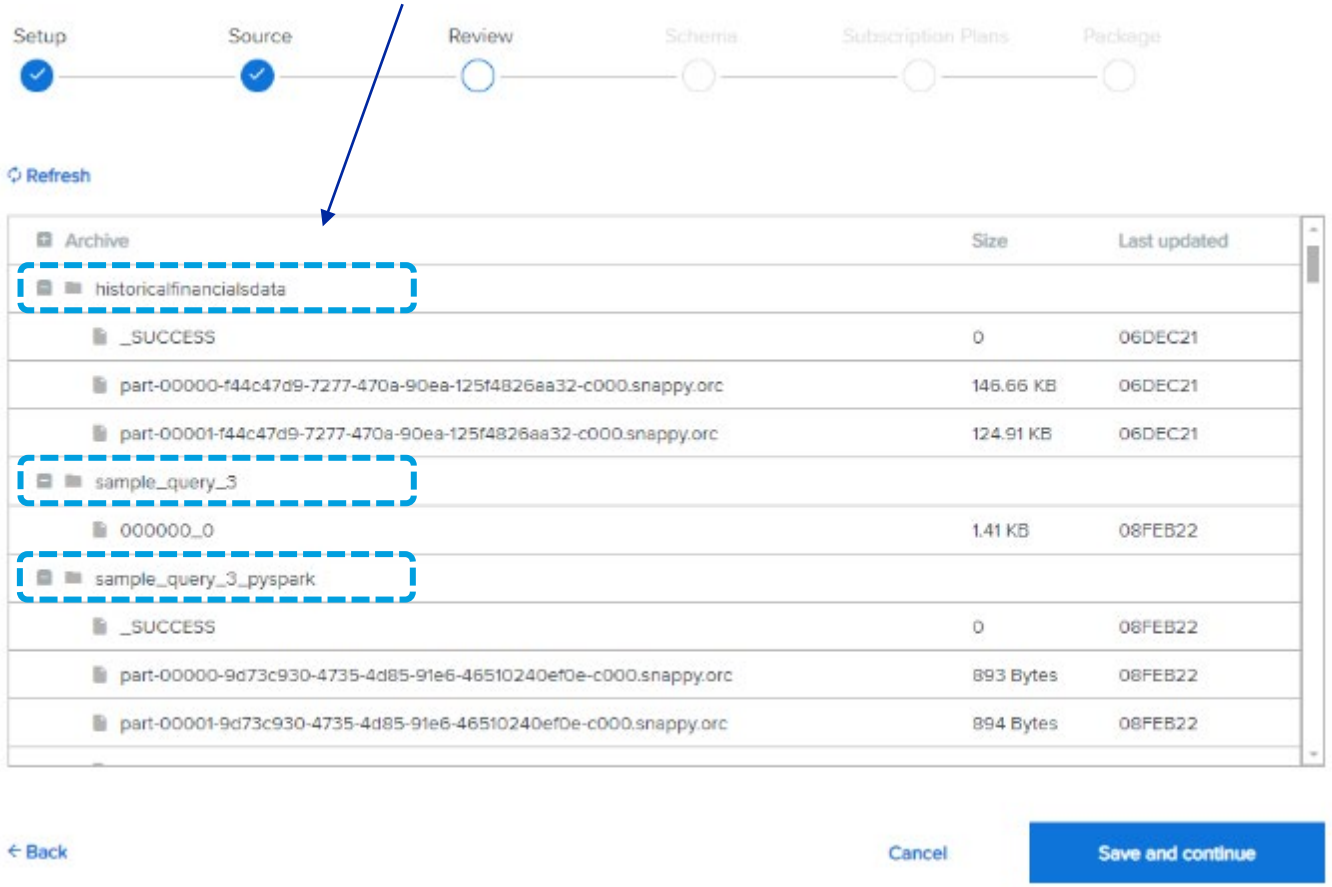
“Engineered data product” is used to publish a dataset that was created in “Spaces”.
It ingests modified data from your ‘Space’ that were saved under the ‘publish_db’ database.



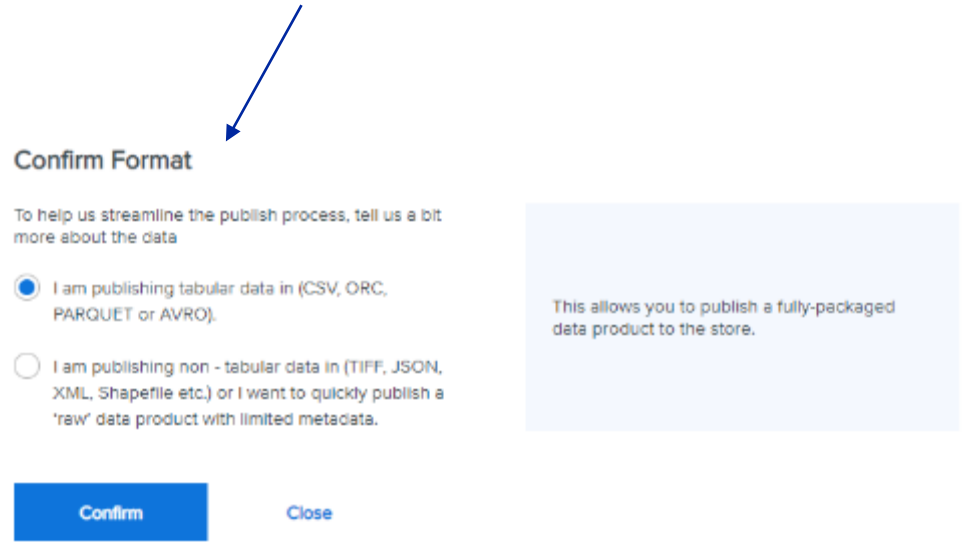
Select the “Space” the modified data is being retrieved from.

Products – Create “Engineered data product” (cont.)

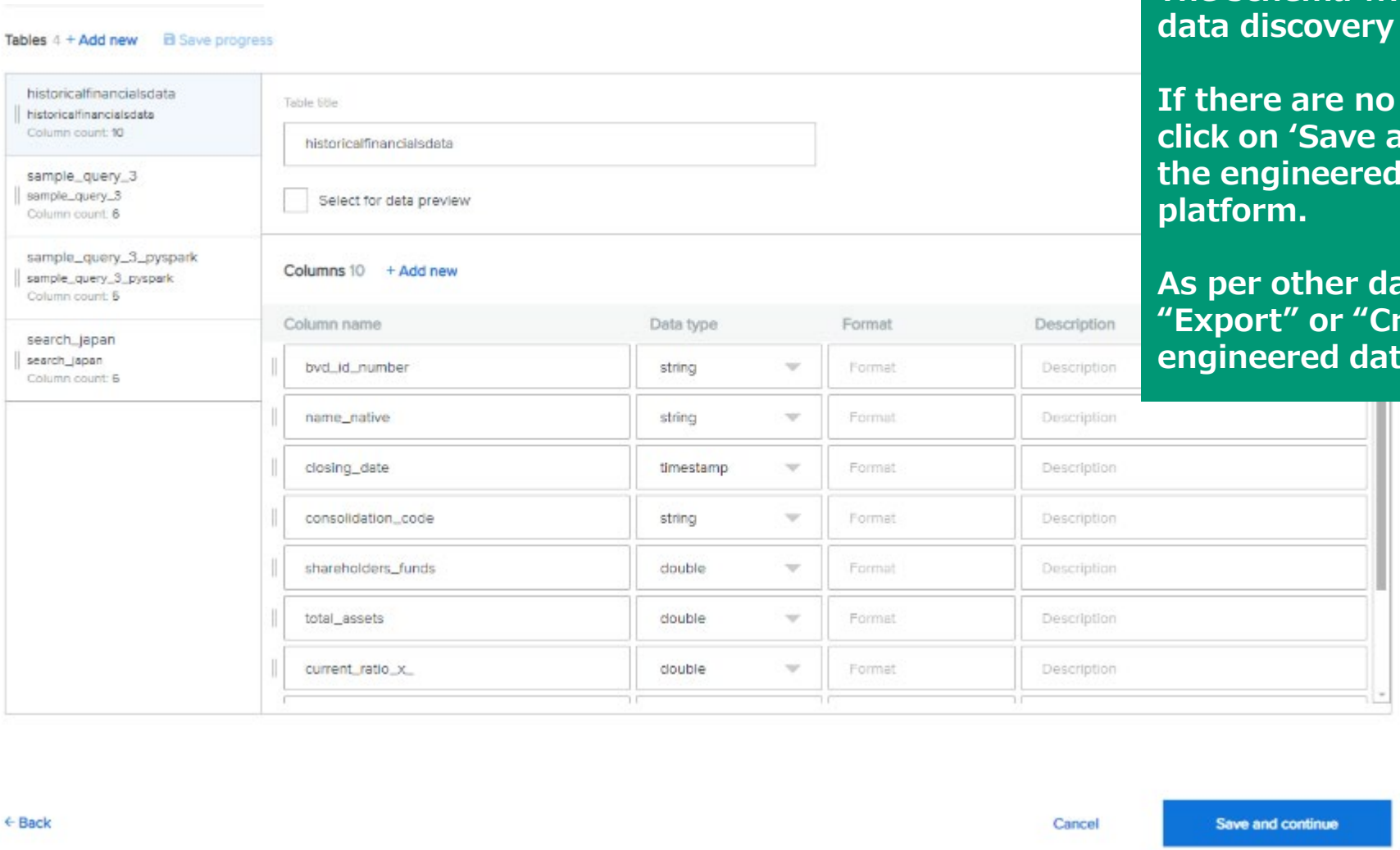
Created tables or datasets that were saved in the “publish_db” database will be shown here.



Confirm the format of the modified data you wish to publish



Products – Create “Engineered data product” (cont.)



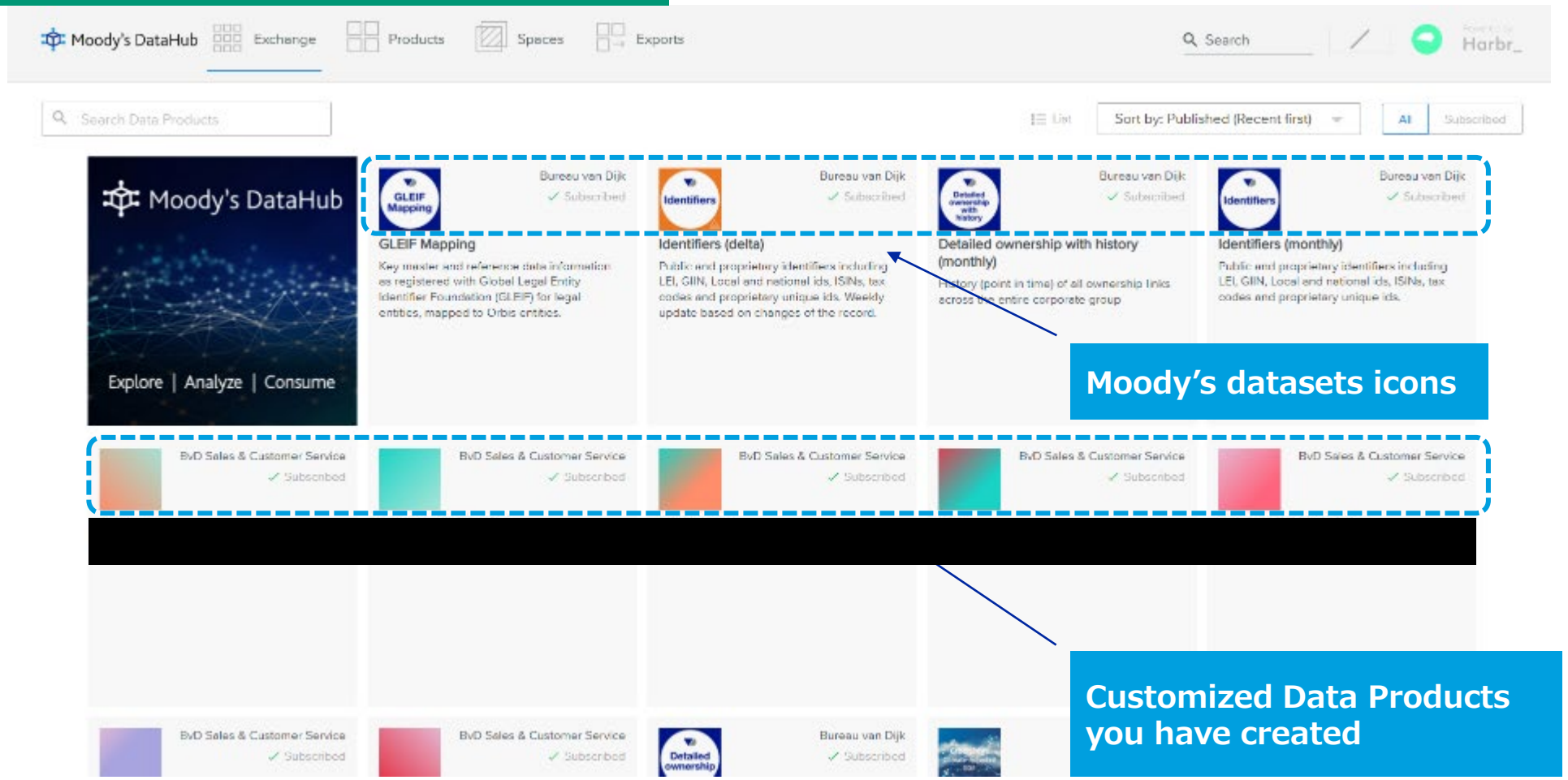
The schema will be presented once the data discovery process is completed.

If there are no issues with the data, click on ‘Save and continue’ to publish the engineered dataset on to the DataHub platform.

As per other datasets, you may choose to “Export” or “Create Space” with this new engineered dataset.

Products – Appearance in “Exchange”

Please note that the Moody’s datasets and the Product that you created will have different icons. Customized datasets will have different colored icons.



6 Sample Queries

Sample Queries

+ amazon WorkSpaces

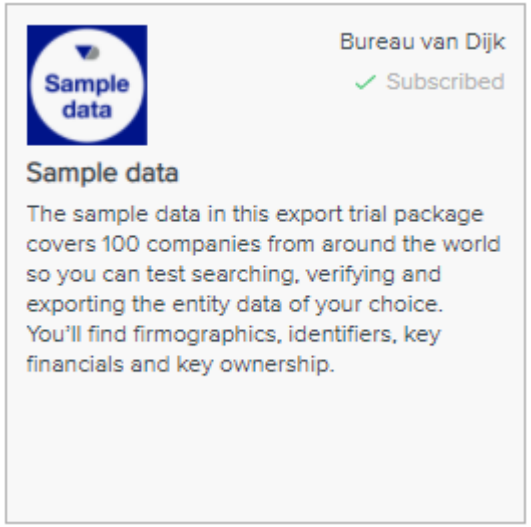
Performing SQL Queries

Amazon Workspaces is required to access a created Space. To perform a specific query on Hue and PySpark, the database name and table is required.

An example database name is db_trial_package__standard_39286a51.

An example of a table name is key_financials.

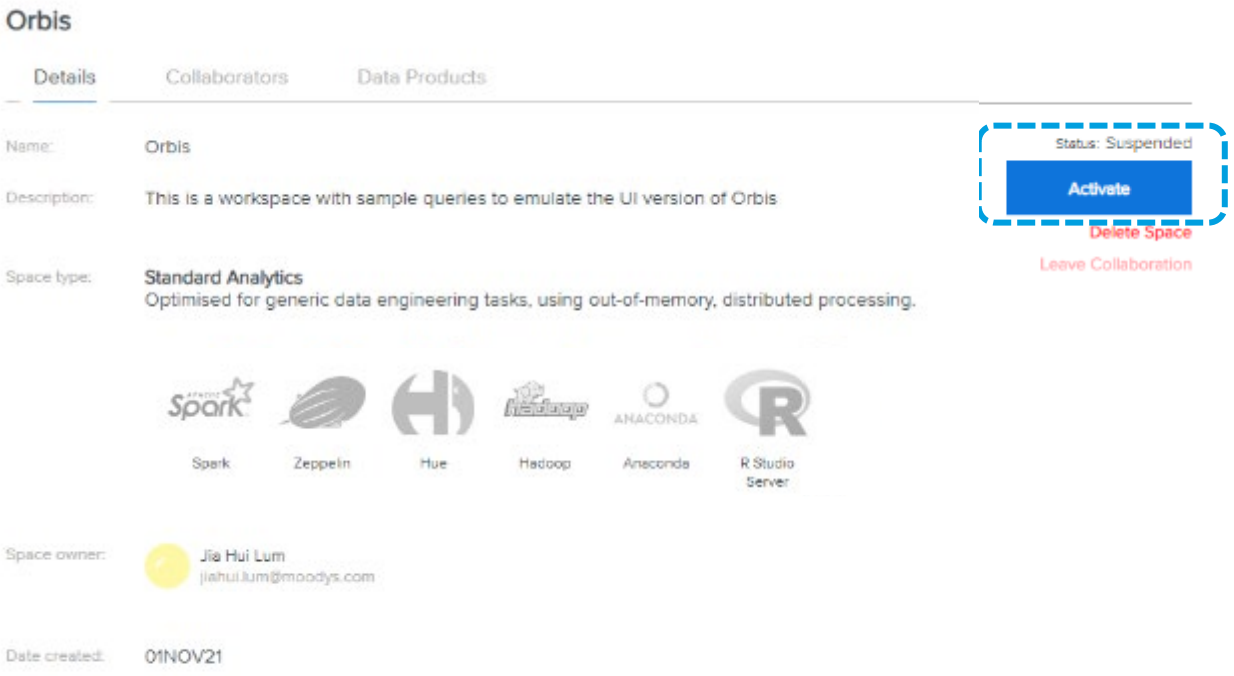
For the following examples, 'Sample data' dataset by BvD will be used.



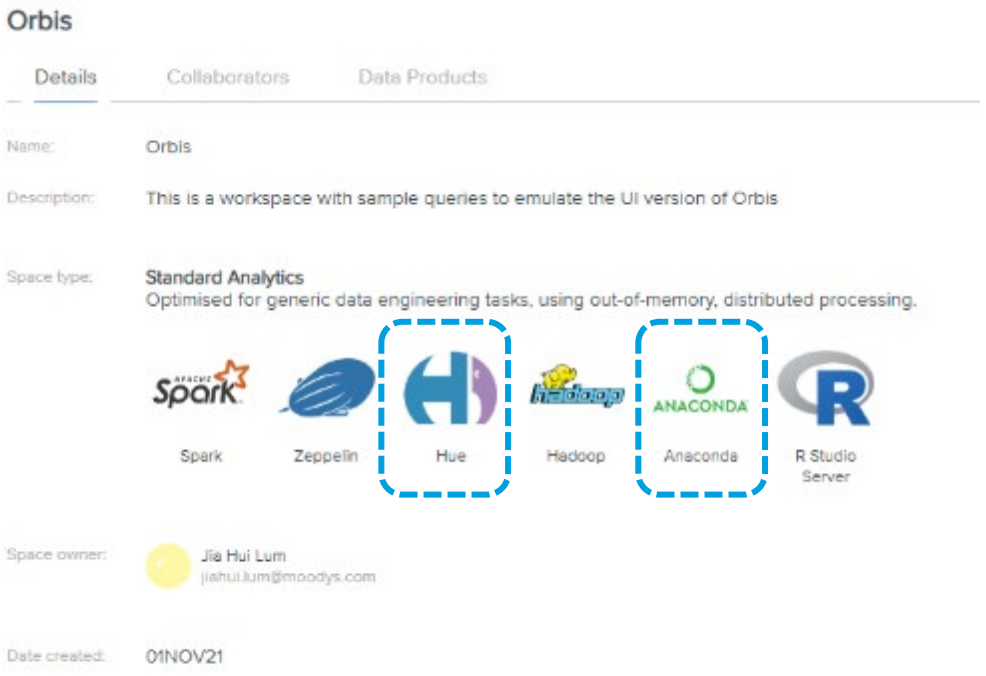
Sample Queries

– Activating a Space –

1. **1. Open Amazon WorkSpaces and login with your credentials to access the cloud platform. Click on the 'Activate' button to initiate the space that you've created.**



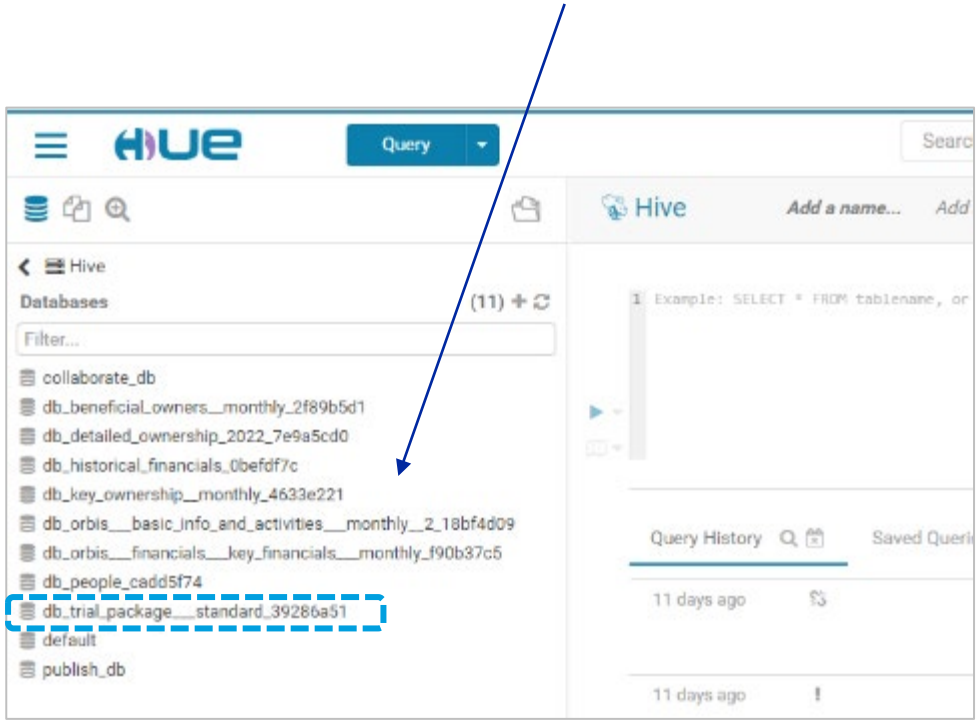
2. **2. Once the space is activated (see FAQs for time taken to activate a space), open *Hue* and *Anaconda (PySpark)* by clicking on it.**

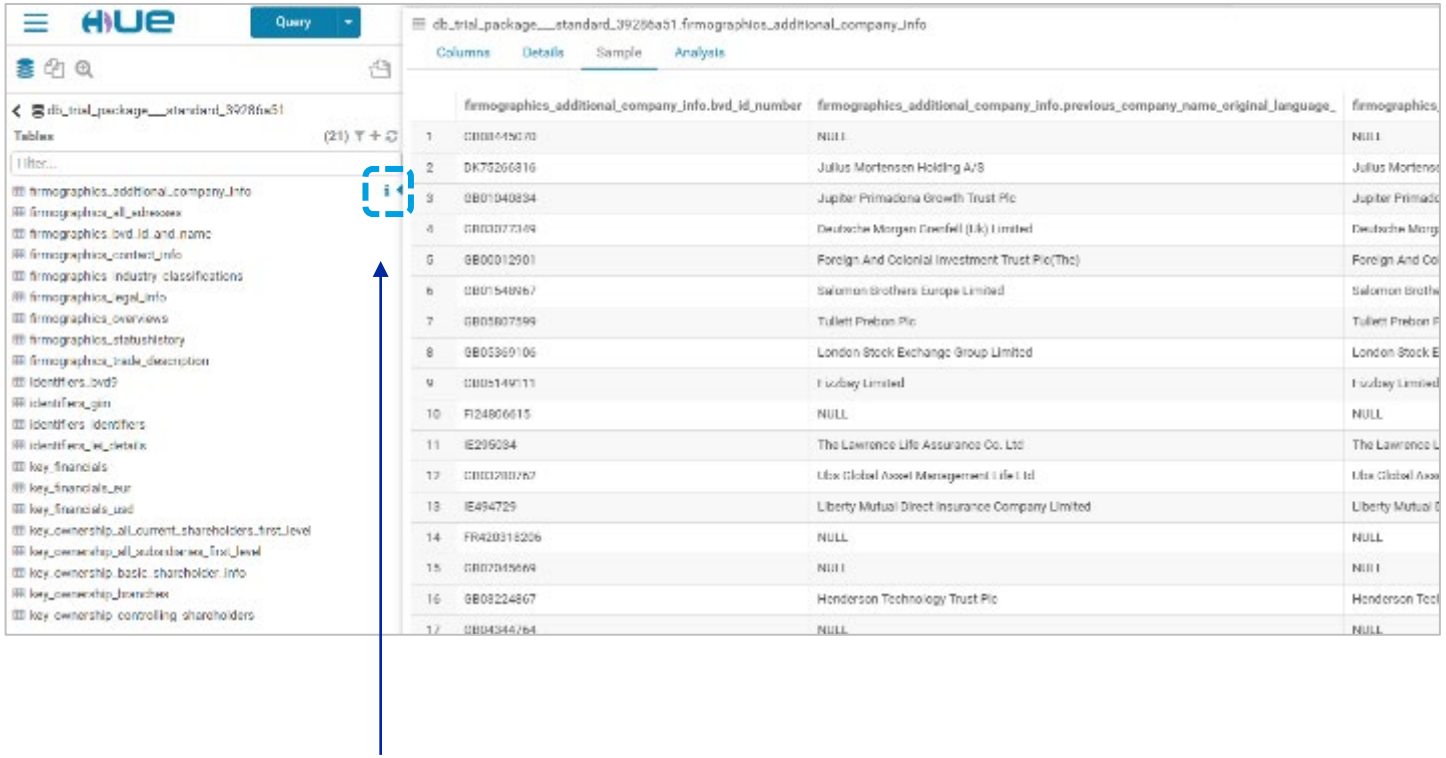


Sample Queries

– Exploring the Dataset using ***Hue*** –

Click the Orbis Sample Data called Database ‘db_trial_package__standard_39286a51’





To explore the tables & sample data within the database, hover over the desired table and click on the ‘i’ button.

Sample Queries

– Exploring the Dataset using ***PySpark*** –

Create a new PySpark notebook to perform the following queries.

To retrieve the database within the Space

dbs_sql = 'show databases'
dbs = sqlContext.sql(dbs_sql)
dbs.toPandas()

```
dbs_sql = 'show databases'
dbs = sqlContext.sql(dbs_sql)
dbs.toPandas()
```

	databaseName
0	collaborate_db
1	db_beneficial_owners__monthly_2f89b5d1
2	db_detailed_ownership_2022_7e9a5cd0
3	db_historical_financials_0befdf7c
4	db_key_ownership__monthly_4633e221
5	db_orbis___basic_info_and_activities___monthly...
6	db_orbis___financials___key_financials___month...
7	db_people_cadd5f74
8	db_trial_package___standard_39286a51
9	default
10	publish_db

To explore the tables & sample data within the database

read_db =
'db_trial_package___standard_39286a51'
tbls_sql = 'show tables in {}'.format(read_db)
tbls = sqlContext.sql(tbls_sql)
tbls.toPandas()

```
read_db = 'db_trial_package___standard_39286a51'
tbls_sql = 'show tables in {}'.format(read_db)
tbls = sqlContext.sql(tbls_sql)
tbls.toPandas()
```

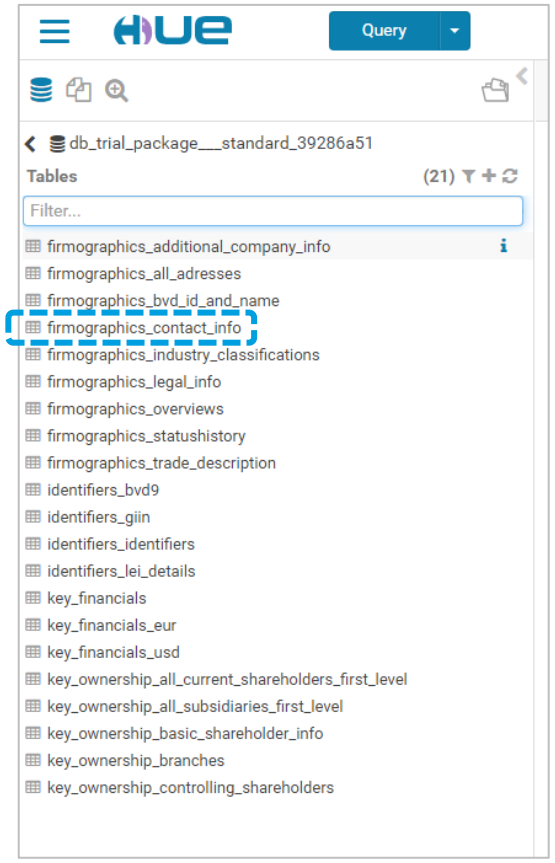
	database	tableName	isTemporary
0	db_trial_package___standard_39286a51	firmographics_additional_company_info	False
1	db_trial_package___standard_39286a51	firmographics_all_adresses	False
2	db_trial_package___standard_39286a51	firmographics_bvd_id_and_name	False
3	db_trial_package___standard_39286a51	firmographics_contact_info	False
4	db_trial_package___standard_39286a51	firmographics_industry_classifications	False
5	db_trial_package___standard_39286a51	firmographics_legal_info	False
6	db_trial_package___standard_39286a51	firmographics_overviews	False
7	db_trial_package___standard_39286a51	firmographics_statushistory	False
8	db_trial_package___standard_39286a51	firmographics_trade_description	False
9	db_trial_package___standard_39286a51	identifiers_bvd9	False
10	db_trial_package___standard_39286a51	identifiers_giin	False
11	db_trial_package___standard_39286a51	identifiers_identifiers	False
12	db_trial_package___standard_39286a51	identifiers_lei_details	False
13	db_trial_package___standard_39286a51	key_financials	False
14	db_trial_package___standard_39286a51	key_financials_eur	False
15	db_trial_package___standard_39286a51	key_financials_usd	False
16	db_trial_package___standard_39286a51	key_ownership_all_current_shareholders_first_l...	False
17	db_trial_package___standard_39286a51	key_ownership_all_subsidiaries_first_level	False
18	db_trial_package___standard_39286a51	key_ownership_basic_shareholder_info	False
19	db_trial_package___standard_39286a51	key_ownership_branches	False
20	db_trial_package___standard_39286a51	key_ownership_controlling_shareholders	False

Sample ***Hue*** Query 1

– Lookup a list of BvD IDs by country –

Learning point 1: How to filter on 1 criterion

*Highlighted in blue requires user inputs



select bvd_id_number, country
from db_trial_package___standard_39286a51.firmographics_contact_info
where firmographics_contact_info.country = '<country>'

```
select bvd_id_number, country
from db_trial_package___standard_39286a51.firmographics_contact_info
where firmographics_contact_info.country = 'United Kingdom'
```

Query History | Saved Queries | Query Builder | Results (51)

	bvd_id_number	country
1	GB03077349	United Kingdom
2	GB01555746	United Kingdom
3	GB05892452	United Kingdom
4	GB03584019	United Kingdom
5	GB01548967	United Kingdom
6	GB01404713	United Kingdom
7	GB00101625	United Kingdom

Sample ***PySpark*** Query 1

– Lookup a list of BvD IDs by country –

Learning point 1: How to filter on 1 criterion

*Highlighted in blue requires user inputs

```
read_db = 'db_trial_package___standard_39286a51'
read_table = 'firmographics_contact_info'
select_sql = 'select * from {}.{}'.format(read_db, read_table)
sprk_df = sqlContext.sql(select_sql)

sprk_df.where(sprk_df.country == '<country>').select('bvd_id_number', 'country').show()
```

```
read_db = 'db_trial_package___standard_39286a51'
read_table = 'firmographics_contact_info'
select_sql = 'select * from {}.{}'.format(read_db, read_table)
sprk_df = sqlContext.sql(select_sql)

sprk_df.where(sprk_df.country == 'United Kingdom').select('bvd_id_number', 'country').show()
```

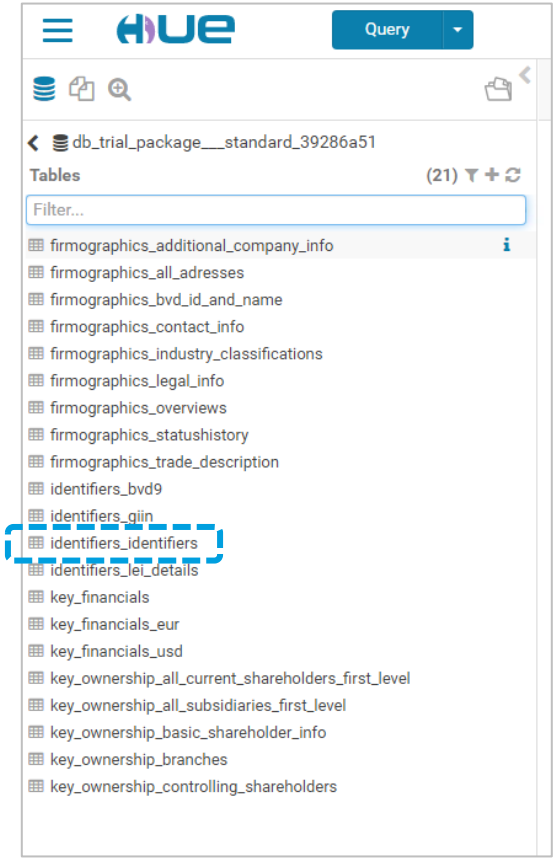
```
+-------------+--------------+
|bvd_id_number|       country|
+-------------+--------------+
|   GB03077349|United Kingdom|
|   GB01555746|United Kingdom|
|   GB05892452|United Kingdom|
|   GB03584019|United Kingdom|
|   GB01548967|United Kingdom|
|   GB01404713|United Kingdom|
|   GB00101625|United Kingdom|
|   GB04344764|United Kingdom|
```

Sample ***Hue*** Query 2

– Lookup a Single BvD ID using identifiers –

Learning point 2: How to filter on multiple criteria

*Highlighted in blue requires user inputs




```
select bvd_id_number, national_id_number, national_id_label
from db_trial_package___standard_39286a51.identifiers_identifiers
where identifiers_identifiers.national_id_label = '<id_type>' and
identifiers_identifiers.national_id_number = '<id_number>'
```

```
1 select bvd_id_number, national_id_number, national_id_label
2 from db_trial_package___standard_39286a51.identifiers_identifiers
3 where identifiers_identifiers.national_id_label = 'LEI' and identifiers_identifiers.national_id_number = '5493009W7D2OFJBGG513'
4
```

Query History | Saved Queries | Query Builder | Results (1)

	bvd_id_number	national_id_number	national_id_label
1	GB04486031	5493009W7D2OFJBGG513	LEI

Sample ***PySpark*** Query 2

– Lookup a Single BvD ID using identifiers –

Learning point 2: How to filter on multiple criteria

*Highlighted in blue requires user inputs

```
read_db = 'db_trial_package___standard_39286a51'
read_table = 'identifiers_identifiers'
select_sql = 'select * from {}.{}'.format(read_db, read_table)
sprk_df = sqlContext.sql(select_sql)

sprk_df.where((sprk_df.national_id_label == '<id_type>') & (sprk_df.national_id_number == '<id_number>'))¥
        .select('bvd_id_number', 'national_id_number', 'national_id_label').collect()
```

```
read_db = 'db_trial_package___standard_39286a51'
read_table = 'identifiers_identifiers'
select_sql = 'select * from {}.{}'.format(read_db, read_table)
sprk_df = sqlContext.sql(select_sql)

sprk_df.where((sprk_df.national_id_label == 'LEI' ) & (sprk_df.national_id_number == '5493009W7D2OFJBGG513'))\
        .select('bvd_id_number', 'national_id_number', 'national_id_label').collect()
```

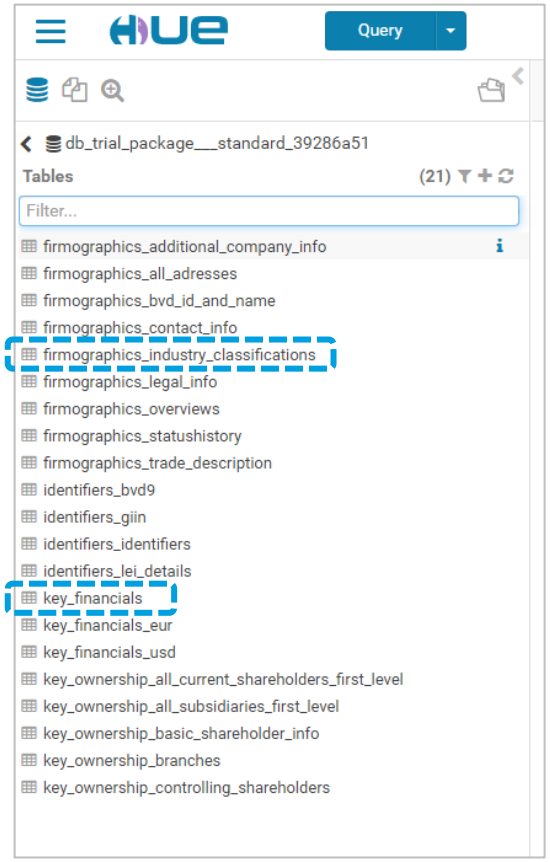
```
[Row(bvd_id_number='GB04486031', national_id_number='5493009W7D2OFJBGG513', national_id_label='LEI')]
```

Sample ***Hue*** Query 3

– Lookup a list of Financials based on industry classification –

Learning point 3: How to join 2 tables together

*Highlighted in blue requires user inputs



select fic.bvd_id_number, fic.naics_core_code_4_digits_, kf.consolidation_code, kf.closing_date, kf.total_assets, kf.cash_flow
from db_trial_package___standard_39286a51.firmographics_industry_classifications fic
inner join db_trial_package___standard_39286a51.key_financials kf on kf.bvd_id_number = fic.bvd_id_number
where fic.naics_core_code_4_digits_ = '<industry_code>'
and kf.consolidation_code = '<consolidation code>'
order by fic.bvd_id_number, kf.closing_date

```
1  select fic.bvd_id_number
2      , fic.naics_core_code_4_digits_
3      , kf.consolidation_code
4      , kf.closing_date
5      , kf.total_assets
6      , kf.cash_flow
7  from db_trial_package___standard_39286a51.firmographics_industry_classifications fic
8  inner join db_trial_package___standard_39286a51.key_financials kf on kf.bvd_id_number = fic.bvd_id_number
9  where fic.naics_core_code_4_digits_ = '5239'
10 and kf.consolidation_code = 'C2'
11 order by fic.bvd_id_number, kf.closing_date
```

	fic.bvd_id_number	fic.naics_core_code_4_digits_	kf.consolidation_code	kf.closing_date	kf.total_assets	kf.cash_flow
1	FR349501676	5239	C2	2006-12-31 00:00:00.0	223747612	76857949
2	FR349501676	5239	C2	2007-12-31 00:00:00.0	334949136	101378390
3	FR349501676	5239	C2	2009-12-31 00:00:00.0	889171572	320253536
4	FR349501676	5239	C2	2010-12-31 00:00:00.0	1270880008	480343695
5	FR349501676	5239	C2	2011-12-31 00:00:00.0	1434152177	287405214
6	FR349501676	5239	C2	2012-12-31 00:00:00.0	1722821403	343162097
7	FR349501676	5239	C2	2013-12-31 00:00:00.0	2029693165	274619912
8	FR349501676	5239	C2	2014-12-31 00:00:00.0	2292263664	286759435

Sample ***PySpark*** Query 3

– Lookup a list of Financials based on industry classification –

Learning point 3: How to join 2 tables together

*Highlighted in blue requires user inputs

```
read_db = 'db_trial_package___standard_39286a51'
read_fic_table = 'firmographics_industry_classifications'
select_sql = 'select bvd_id_number, naics_core_code_4_digits_ from {}.{}'.format(read_db, read_fic_table)
sprk_fic_df = sqlContext.sql(select_sql)

read_kf_table = 'key_financials'
select_sql = 'select bvd_id_number, consolidation_code, closing_date, total_assets \
           , cash_flow from {}.{}'.format(read_db, read_kf_table)
sprk_kf_df = sqlContext.sql(select_sql)

combined = sprk_fic_df.filter(sprk_fic_df.naics_core_code_4_digits_ == '5239')\
                  .join(sprk_kf_df.filter(sprk_kf_df.consolidation_code == 'C2'), 'bvd_id_number', 'inner')
combined.sort('bvd_id_number', 'closing_date').show()
```

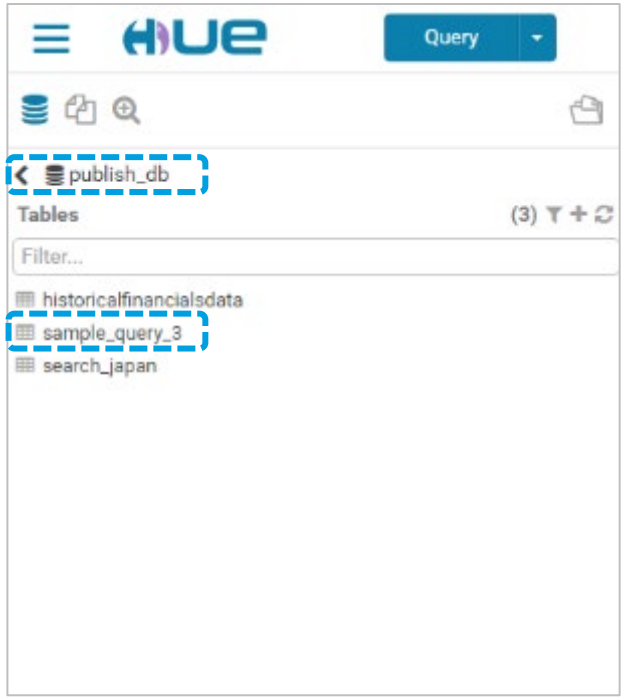
bvd_id_number	naics_core_code_4_digits_	consolidation_code	closing_date	total_assets	cash_flow
FR349501676	5239	C2	2006-12-31 00:00:00	2.23747612E8	7.6857049E7
FR349501676	5239	C2	2007-12-31 00:00:00	3.34949138E8	1.0137839E8
FR349501676	5239	C2	2009-12-31 00:00:00	8.89171572E8	3.20253536E8
FR349501676	5239	C2	2010-12-31 00:00:00	1.270880008E9	4.80343695E8
FR349501676	5239	C2	2011-12-31 00:00:00	1.434152177E9	2.87485214E8
FR349501676	5239	C2	2012-12-31 00:00:00	1.722821403E9	3.4316209 7E8
FR349501676	5239	C2	2013-12-31 00:00:00	2.029693165E9	2.74619912E8
FR349501676	5239	C2	2014-12-31 00:00:00	2.292263664E9	2.8075043 5E8
FR349501676	5239	C2	2016-12-31 00:00:00	2.858641541E9	2.1921489 6E8
FR349501676	5239	C2	2017-12-31 00:00:00	3.026127861E9	1.7796902 8E8
GB00012901	5239	C2	1996-12-31 00:00:00	1.974651E9	null
GB00026351	5239	C2	2010-10-31 00:00:00	5.41802E8	null
GB00026351	5239	C2	2011-10-31 00:00:00	5.24895E8	null
GB00026351	5239	C2	2012-10-31 00:00:00	5.53292E8	null
GB00101625	5239	C2	2009-12-31 00:00:00	1.138805E9	null
GB00101625	5239	C2	2010-12-31 00:00:00	1.261171E9	null
GB00101625	5239	C2	2011-12-31 00:00:00	1.124089E9	null
GB00101625	5239	C2	2012-12-31 00:00:00	1.243845E9	null
GB00101625	5239	C2	2013-12-31 00:00:00	1.501189E9	null

Sample ***Hue*** Query 4

– Save query results into a new table –

Learning point 4: How to publish and export results

* Sample query 3 is used and saved into a new table in publish_db database
  This step is required to publish an engineered data product (see pages 47-49)
  Highlighted in blue requires user inputs



drop table if exists publish_db.sample_query_3; create table if not exists publish_db.sample_query_3 as
select fic.bvd_id_number, fic.naics_core_code_4_digits_, kf.consolidation_code, kf.closing_date, kf.total_assets, kf.cash_flow
from db_trial_package___standard_39286a51.firmographics_industry_classifications fic
inner join db_trial_package___standard_39286a51.key_financials kf on kf.bvd_id_number = fic.bvd_id_number
where fic.naics_core_code_4_digits_ = '<industry_code>'
and kf.consolidation_code = '<consolidation code>'
order by fic.bvd_id_number, kf.closing_date

```
1 DROP TABLE IF EXISTS publish_db.sample_query_3; CREATE TABLE IF NOT EXISTS publish_db.sample_query_3 AS
2 select fic.bvd_id_number, fic.naics_core_code_4_digits_, kf.consolidation_code, kf.closing_date, kf.total_assets, kf.cash_flow
3 from db_trial_package___standard_39286a51.firmographics_industry_classifications fic
4 inner join db_trial_package___standard_39286a51.key_financials kf on kf.bvd_id_number = fic.bvd_id_number
5 where fic.naics_core_code_4_digits_ = '5239'
6 and kf.consolidation_code = 'C2'
7 order by fic.bvd_id_number, kf.closing_date
8
```

Sample ***PySpark*** Query 4

– Save query results into a new table –

Learning point 4: How to publish and export results

* Sample query 3 is used and saved into a new table in publish_db database
  This step is required to publish an engineered data product (see pages 47-49)
  Highlighted in blue requires user inputs

```
read_db = 'db_trial_package___standard_39286a51'
read_fic_table = 'firmographics_industry_classifications'
select_sql = 'select bvd_id_number, naics_core_code_4_digits_ from {}.{}'.format(read_db, read_fic_table)
sprk_fic_df = sqlContext.sql(select_sql)

read_kf_table = 'key_financials'
select_sql = 'select bvd_id_number, consolidation_code, closing_date, total_assets from {}.{}'.format(read_db, read_kf_table)
sprk_kf_df = sqlContext.sql(select_sql)

combined = sprk_fic_df.where(sprk_fic_df.naics_core_code_4_digits_ == '5239').\
           join(sprk_kf_df.filter(sprk_kf_df.consolidation_code == 'C2'), 'bvd_id_number', 'inner')

df = combined.toPandas()
df_spark_test = sqlContext.createDataFrame(df)
df_spark_test.printSchema()
df_spark_test.write.format('orc').mode('overwrite').saveAsTable('{}.sample_query_3_pyspark'.format('publish_db'))
```

```
root
 |-- bvd_id_number: string (nullable = true)
 |-- naics_core_code_4_digits_: string (nullable = true)
 |-- consolidation_code: string (nullable = true)
 |-- closing_date: timestamp (nullable = true)
 |-- total_assets: double (nullable = true)
```

To explore the tables & sample data within the 'publish_db' database

```
read_db = 'publish_db'
tbls_sql = 'show tables in {}'.format(read_db)
tbls = sqlContext.sql(tbls_sql)
tbls.toPandas()
```

	database	tableName	isTemporary
0	publish_db	historicalfinancialsdata	False
1	publish_db	sample_query_3	False
2	publish_db	sample_query_3_pyspark	False
3	publish_db	search_japan	False

7

FAQ

1. Username and Password for ***Hue*** & ***RStudio***

The username for Hue is hadoop

The username for Rstudio is rstudio and password is rstudio

2. Will the number of users affect the latency of Spaces? How long does it take to start up spaces?

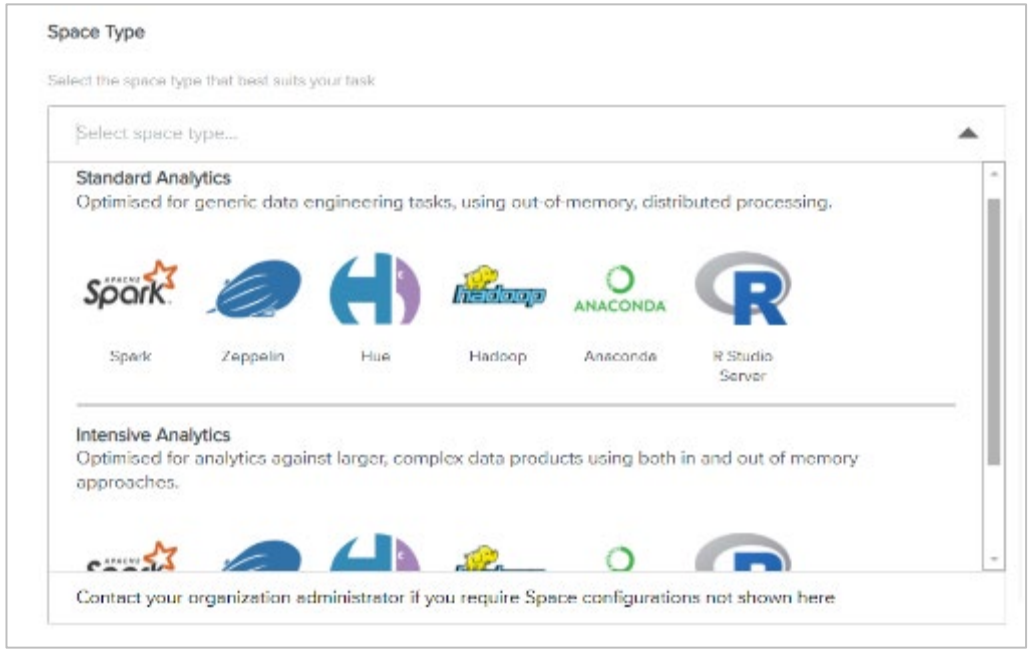
Each space is unique to the user so it should not run any slower.
The size of the Space will impact the computational performance but shouldn't impact the latency in the Space.

We have a pool of clusters active ready to provisioned by a user. If we start a Space from the pool, the startup of your Space will be around 2-4mins. If you do not start your space from the pool, then you will need to wait around 10-15mins for the Space to be available.
The cluster pool in the Moody's DataHub starts at 08:00 UTC and ends at 23:00 UTC.

3. What's the difference between Standard and Intensive space?

A standard Space is comprised of 1 master node and 3 slave nodes, each with 16GB of Memory and 32GB of disk storage.

An intensive Space is made of 1 master and 6 slave nodes, for double the memory usage. Typically, assuming no data skew and efficient programming, an intensive cluster should in all respects execute queries twice as fast as the standard Space.



4. Best practices to export large data products?

- We recommend using an Endpoint for Exporting. It is significantly better than downloading large data to desktop.
- Splitting the product into multiple files. In formats like AVRO, ORC and Parquet, this happens automatically. Generally, if possible, avoid CSV for large exports.
- Individual files shouldn’t be that large. i.e. Having 100 files where 1 file is 90GB and the rest of a fraction of 10GB is not recommended. The parallel copy would chomp through the 99 files in a few minutes and the 90GB one would be left processing for long. Multiple tables isn’t an issue as each table has its own file structure.

Moody's Analytics

Better Faster Decisions

Client Service team
help@bvdinfo.com
+81-3-5408-4288

Moody's Analytics Japan / BvD
+81-3-5408-4280

moodysanalytics.com