Couchbase

Couchbase Workshop

Lab Handbook

DAY 1

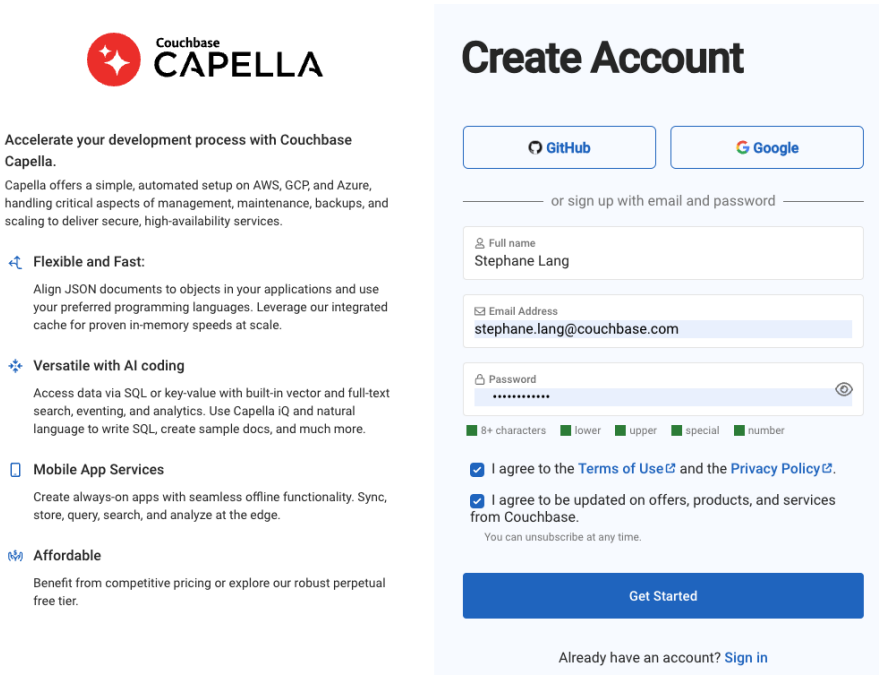
Table of Contents

Lab 1. Get Started

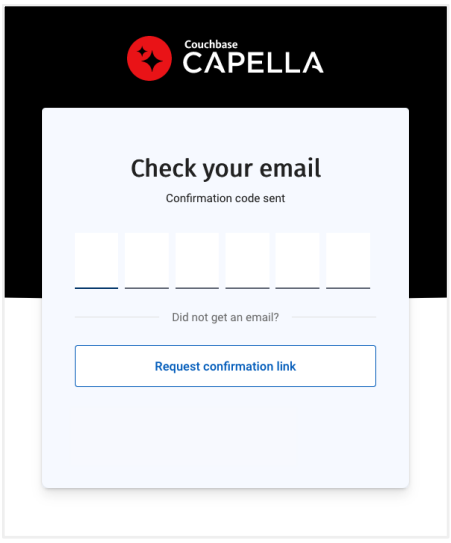
OBJECTIVE: Create a free Capella account and deploy your first database.

1. Go to : https://cloud.couchbase.com/sign-up

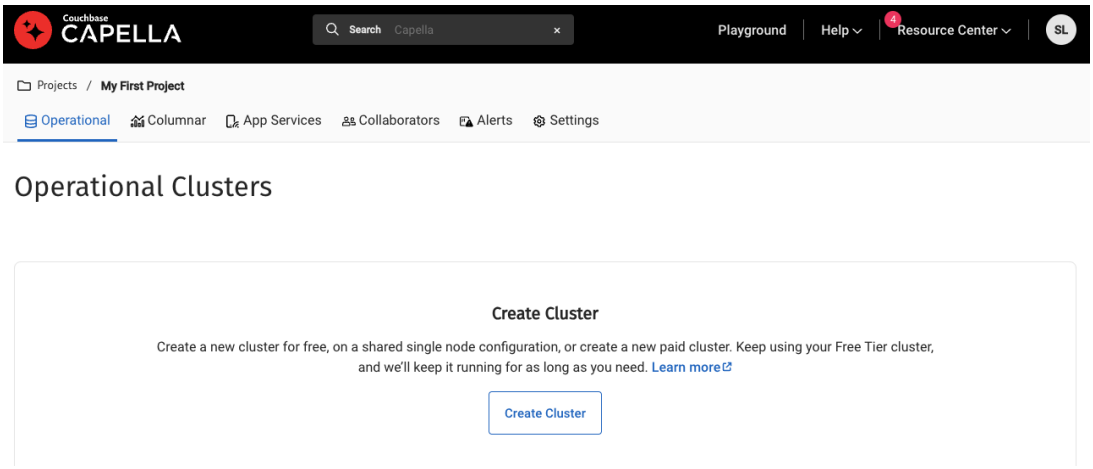
2. On the Sign-up page, enter your name, a corporate email address and a password that meets the security requirements. Check the boxes and click "**Get Started**"



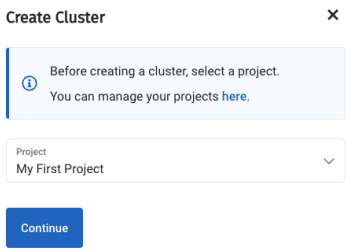
3. Confirm your email address by entering in the code that should be in your inbox.



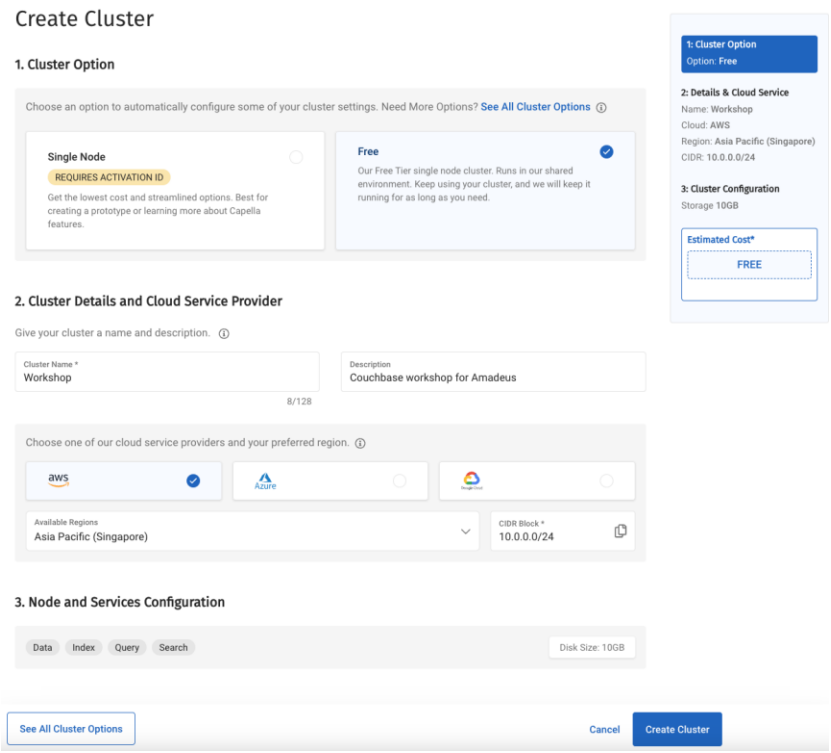
4. Welcome into your free Capella account. While it is a limited in terms of deployment options, services and capabilities, you are able to deploy a cluster for free that will serve your applications. Note that a free account can always be upgraded at any time to enjoy the full capabilities of Couchbase Capella.



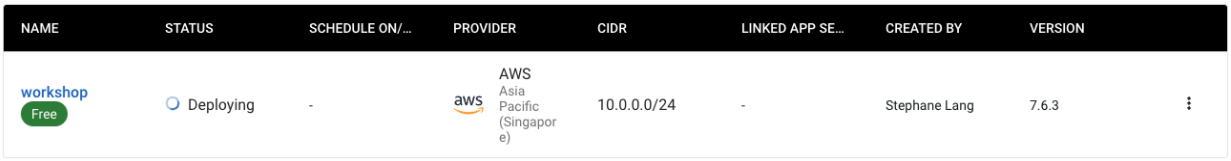
5. Let's create our first cluster. You should be on the "**Operational**" page of the menu. Click on "**Create Cluster**" and select the project "**My First Project**".



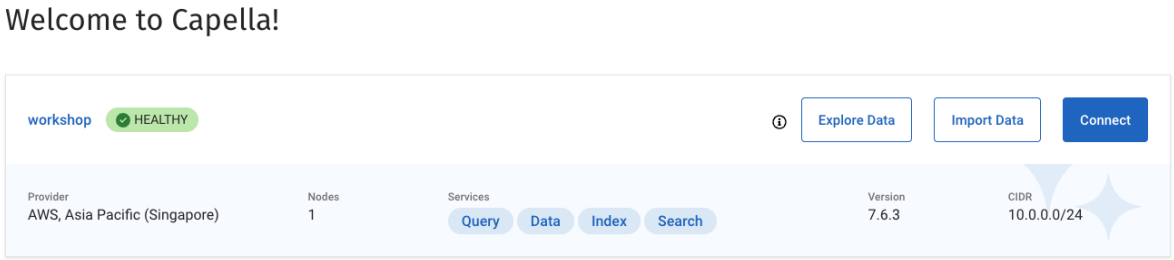
6. This is the cluster creation screen.
   - Choose the category "**Free**" cluster
   - Give it name of your choice, for instance "**workshop**"
   - Select "**AWS**" as the Cloud Service Provider
   - Select the region "**Asia Pacific (Singapore)**"
   - Leave other options as default
   - Finally, click on "**Create Cluster**".



7. The deployment of your Capella Database will take 2 to 3 minutes.



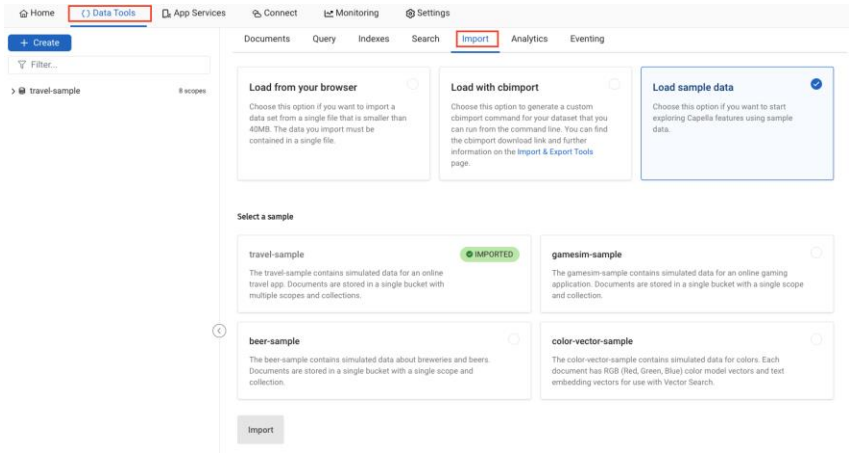
8. Once deployed, you should see your database called “**workshop**” (or whatever you named it) that is available and ready to explore.



9. Explore the Capella user interface. It is dynamic, so menus change depending on where you are in the interface (at the organization level or project level or database level). We may not

cover every capability and service of Capella in this workshop. Feel free to ask questions if you want to know more about a specific item.

10. The free cluster comes with a pre-loaded dataset "**travel-sample**". Other datasets are available, and you can import your own data.
    In the "**Operational**" menu, click on your database name, then go to "**Data Tools**" to see some of the tools and utilities available to interact with the documents stored in the database.
    Click on the "**Import**" sub-menu and select "**Load sample data**" to see the different datasets available. The "**travel-sample**" should be already imported. If not, select it and click on "**Import**".



11. From the import menu, you could also load a file from your computer via the browser (which is slow and recommended for small files only, less than 40MB). The next labs will use the "**travel-sample**" dataset only, so importing another dataset is not required.

Congratulations, you have successfully prepared your first Couchbase Capella environment. You have now a Couchbase cluster up and running with data available. The next step will be to connect an application and read/write from this cluster.

Lab 2. Connect an Application

OBJECTIVE: Review security mechanisms to go through in order to enable an application to access a Capella cluster.

Note: For the sake of convenience of the workshop, every participant will connect to the same application server. The URL or IP address of the application server is provided in class by the instructor. Afterwards, you can follow the same instructions below to connect any server, including your own.

A. Connect to the application server

1. The application that we will run against the Capella cluster is a utility named *cbc-pillowfight*. It comes with every distribution of Couchbase Server, and it is used to generate a workload of reads and writes. For this workshop, we will run the utility from a server in AWS.
   To access the server, you need the access key file `cb-singapore.pem` that your instructor will provide.
   Get the file `cb-singapore.pem` and copy it onto your local laptop. We recommend putting that file it in the `/tmp` folder.

2. Go to the folder that contains the key file and set proper permissions to this file:

   ```
   chmod 400 cb-singapore.pem
   ```

3. Make sure you are still in the folder that contains the access key file and run this command to connect to the AWS server

   ```
   ssh -i "cb-singapore.pem" ec2-user@ec2-url
   ```

   Where *ec2-url* is the URL or the IP address of the application server that your instructor will communicate in class.

4. Enter "**yes**" when you are asked to continue connecting. Your terminal should look like that :

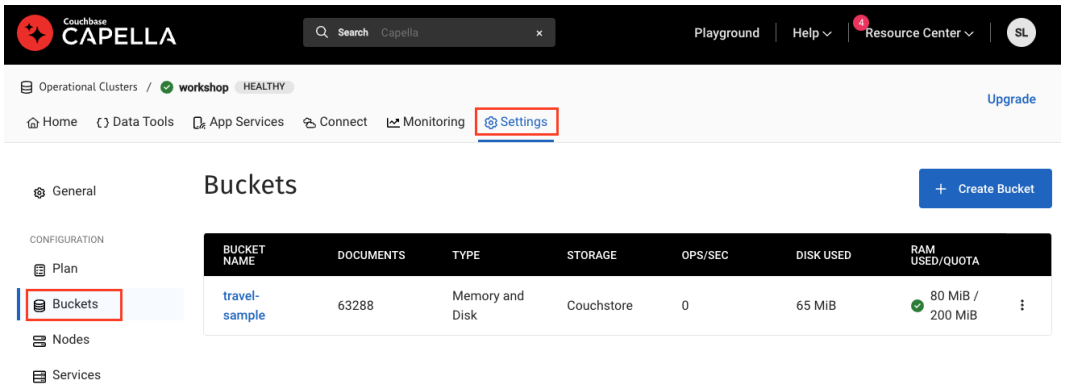
```
ssh -i "cb-singapore.pem" ec2-user@ec2-47-129-211-67.ap-southeast-1.compute.amazonaws.com

The authenticity of host 'ec2-47-129-211-67.ap-southeast-1.compute.amazonaws.com (47.129.211.67)' can't be established.
ED25519 key fingerprint is SHA256:hzSgJmhVKAtmNNeUWjNKPpfk7Eo7Dm1YEwxC37RIgtw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:405: ec2-18-136-202-170.ap-southeast-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes
Warning: Permanently added 'ec2-47-129-211-67.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Mon Aug 26 12:23:38 2024 from 109.221.162.101
```

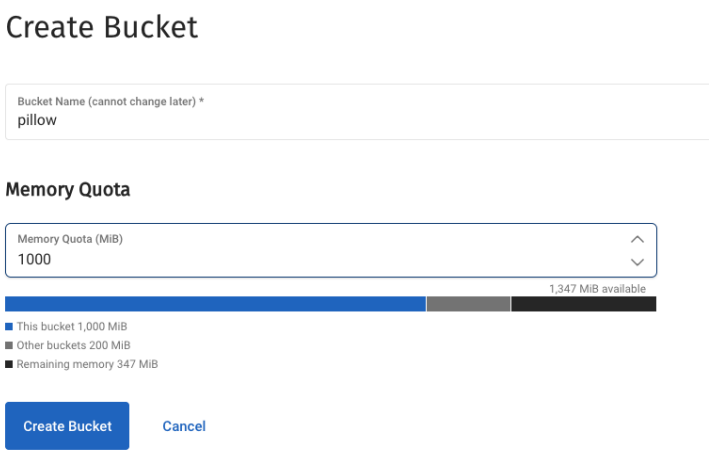
Keep the terminal window open, we will use it later in this lab.

B. Create the destination bucket

5. Go into your cluster interface and click the "**Settings**" menu. Then go to "**Buckets**"

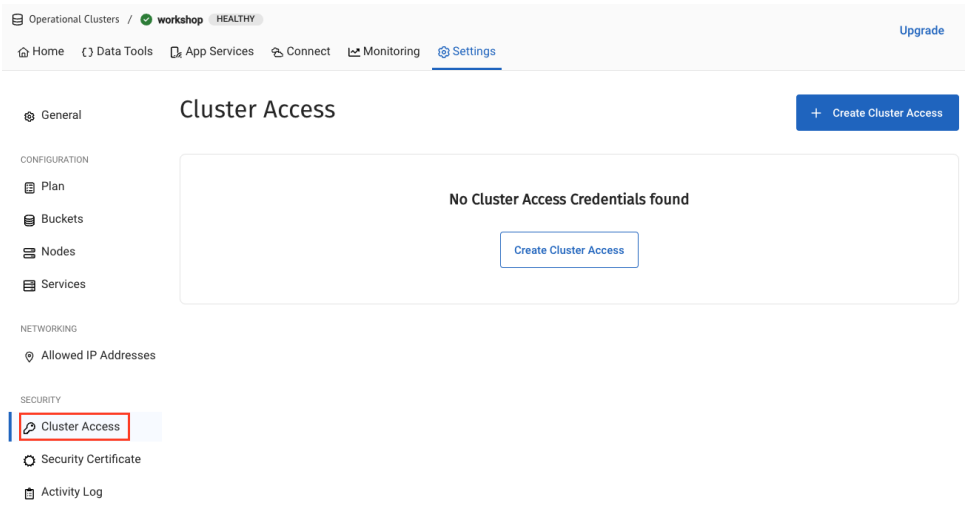


6. Click on "**Create Bucket**" to create a new bucket. Choose "**pillow**" as the bucket name and "**1000**" as the memory quota. Then click on "**Create bucket**"

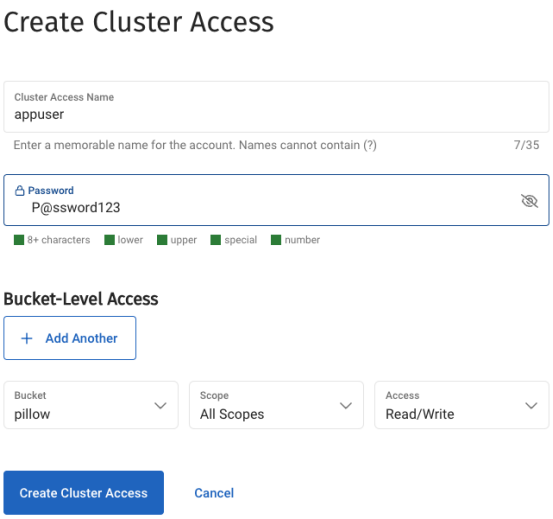


C. Create database credentials

7. Go to "**Cluster Access**" and click the "**Create Cluster Access**" button.



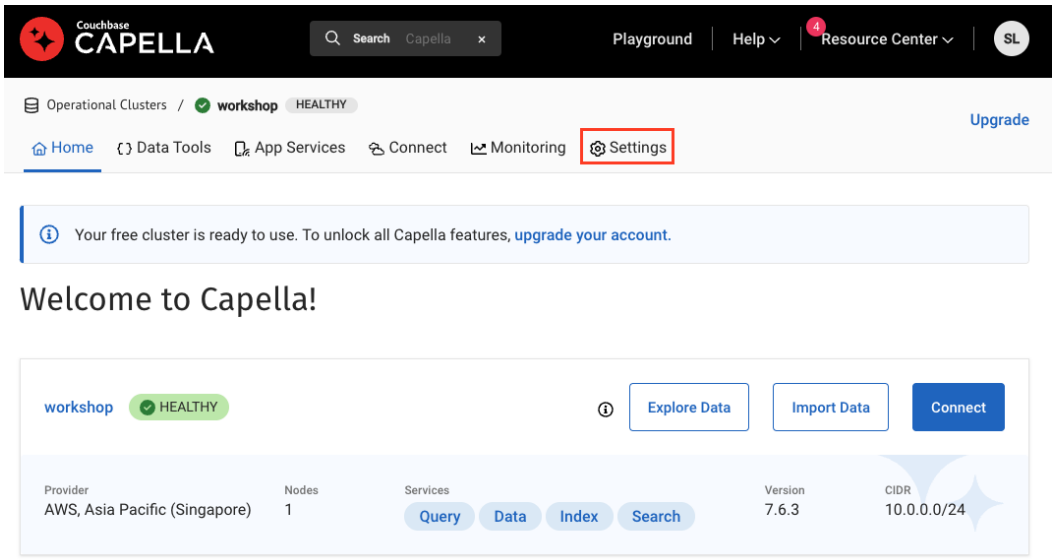
8. Choose “**appuser**” as the username and “**P@ssword123**” as the password. Alternatively, you may choose a username and password of your own, but make sure to remember it as we will need those credentials later in the labs.
   Then, select bucket “**pillow**”, scope “**All scopes**” and “**Read/Write**” access to allow the database user to read and write documents from/into the pillow bucket. Finally, click on “**Create Cluster Access**”



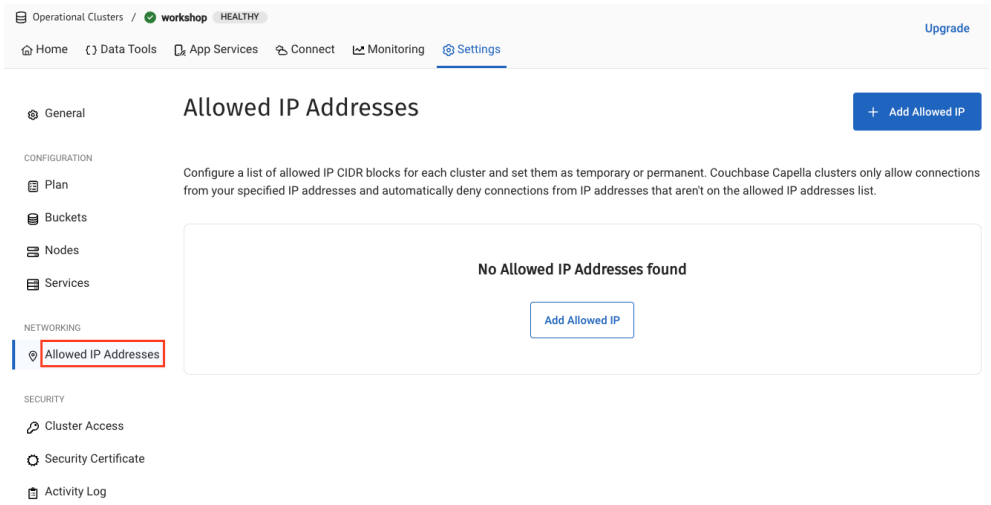
9. You should see the user listed in the interface.

D. <u>Allow IP address</u>

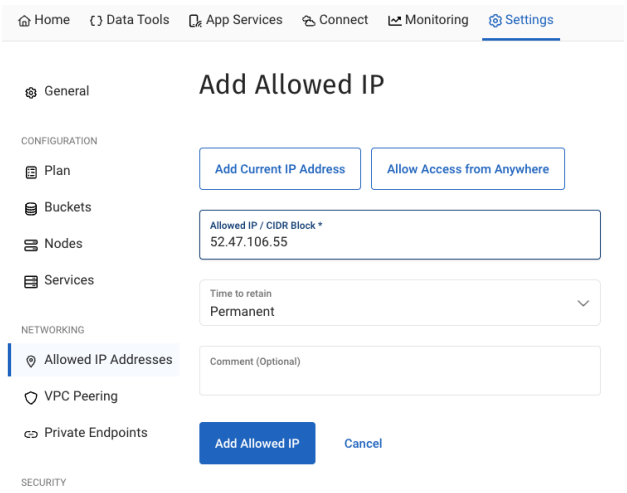
10. Go back into the “**Settings**” of your cluster.



11. Go to “**Allowed IP addresses**” and click the “**Add Allowed IP**” button



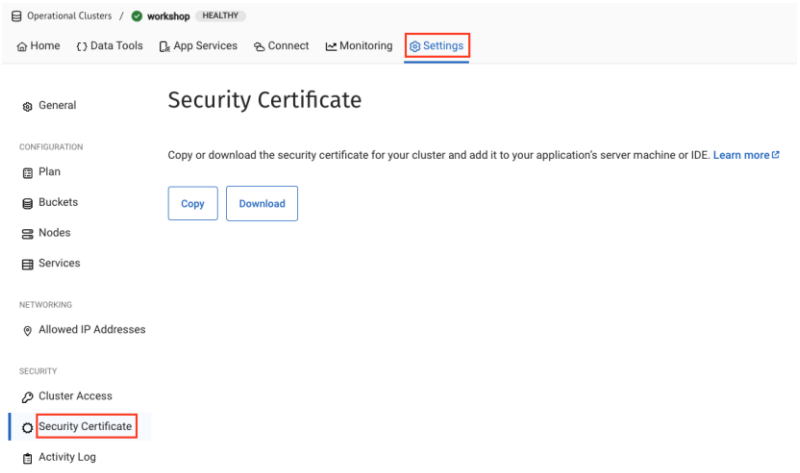
12. Add the IP address provided by your instructor and choose “**Permanent**” to keep it permanently. If you are unsure of which IP address to add, you can also, just for this exercise authorize every IP to access your cluster, by clicking on “*Allow Access from Anywhere*”.



13. You should see the IP listed in the allowed IP addresses.

E. Get cluster certificate

14. Go to the Capella interface, in the "**Settings**" menu, then, go to "**Security Certificate**". This is where you can get the cluster certificate which is required for secured connection. Click on "**Copy**" to copy this certificate, you will need to paste it on the application server.



15. Back to your terminal window, go to the `/tmp` directory of the application server and create a file named "`cert-`*`yourname`*`.pem`", replacing *`yourname`* with your actual name to make it unique. As every participant will connect to the same application server, everyone must create a file containing the certificate of his own cluster to be able to access it.

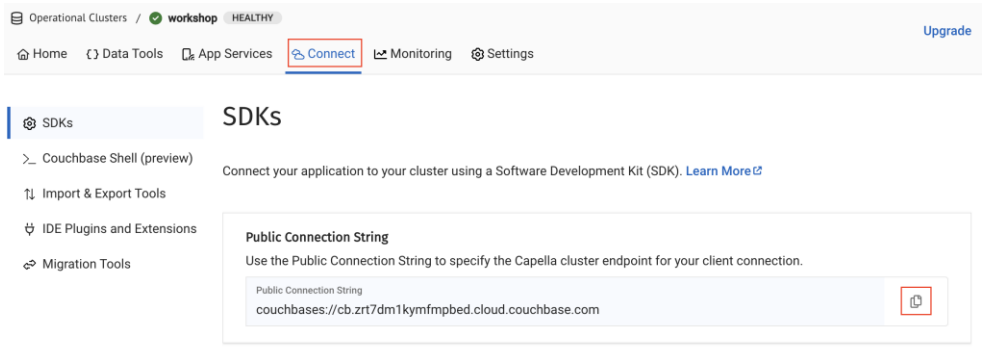
In this file, paste the certificate of the Capella cluster, and save your changes.

F. Cluster URL and run the utility

16. Still in the terminal window, go to the folder `/opt/couchbase/bin` of the AWS server

    ```
    cd /opt/couchbase/bin
    ```

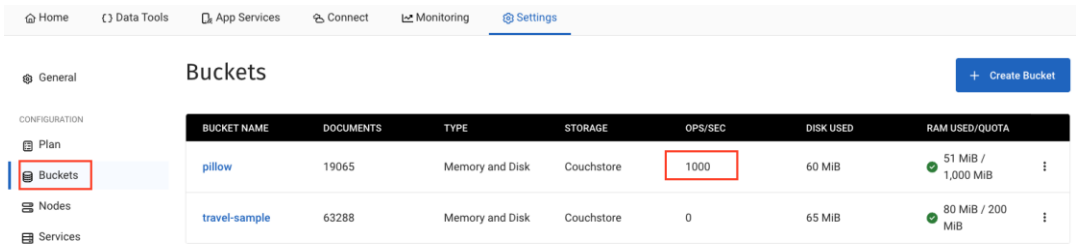
17. Get your unique Capella cluster URL which is displayed in the "**Connect**" menu of you cluster interface. Copy the Public connection string by clicking on the icon next to it:



18. Update this command line with your specific connection string, as well as your specific certificate file name and execute it from the terminal window.

    ```
    ./cbc-pillowfight -U couchbases://xxxxxxxxxxxxx.cloud.couchbase.com/pillow -u appuser -P P@ssword123 --certpath /tmp/cert-xxxxxxxx.pem --json --num-items 200000 --rate-limit 1000
    ```

19. If you obtain an error, please inform your instructor. Otherwise, you should see some activity in your cluster, in the "**Settings**" → "**Buckets**" section.
    You can also check the "**Monitoring**" section to observe the charts available.



20. Then you can stop the utility with Ctrl + C. You may also close the terminal window as we won't need to connect to the application server anymore.

Congratulations, you have successfully connected an application to your Capella cluster. You are now able to enjoy the speed and capabilities of Couchbase into your client application.

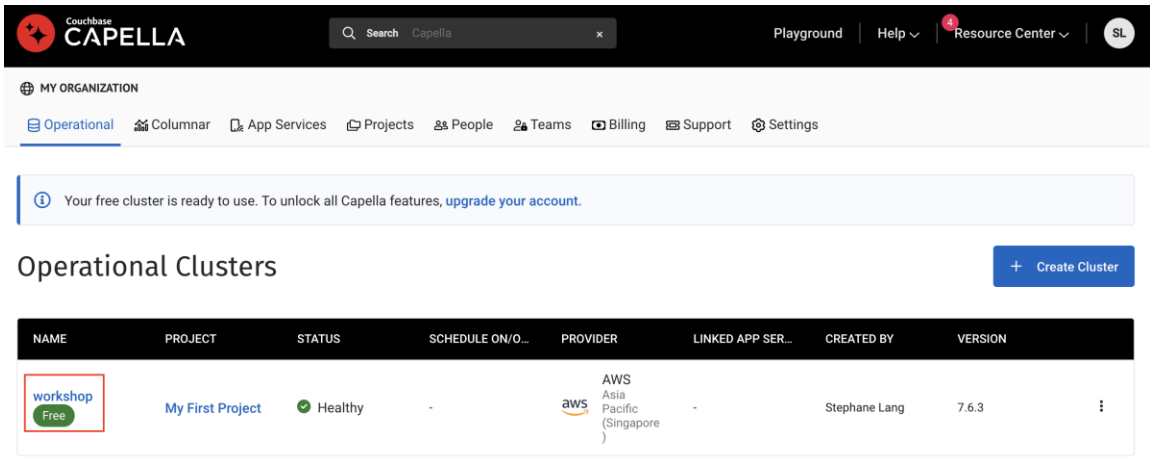
Lab 3. Read from the database

OBJECTIVE: Understand how data is stored, organized and queried in Capella, through SQL++ and Full-text Search.

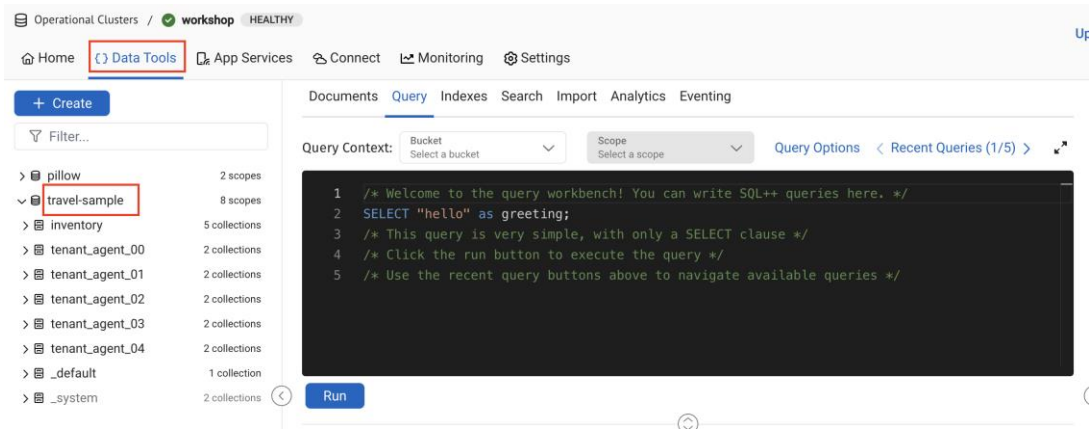
Reminder: In this lab, we will explore one of the default dataset which is "**travel-sample**".

A. SQL++ Queries

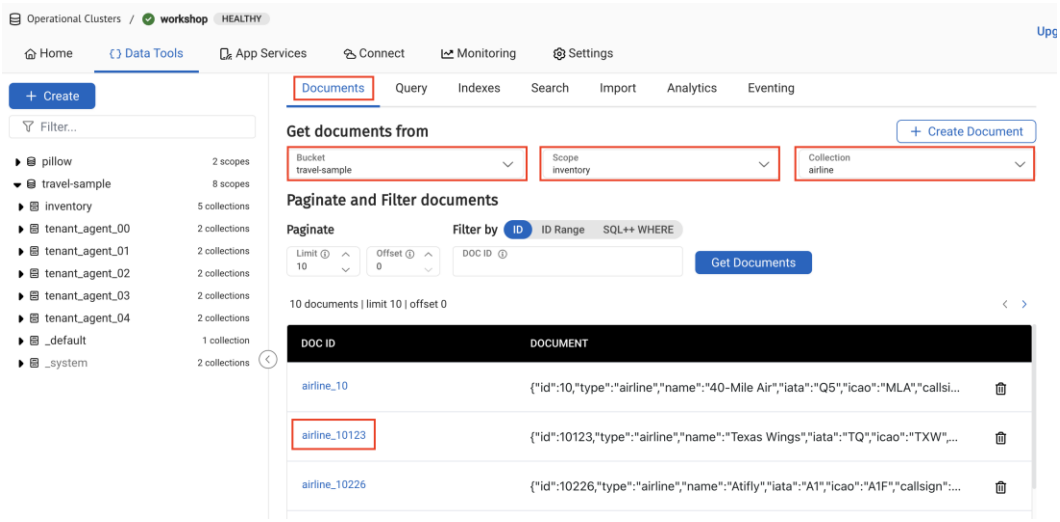
1. From the root screen (click on the Capella logo on the top left corner), click on "**Operational**" and then your cluster name to enter the database user interface.



2. Go to the "**Data Tools**" section, then click on the "**travel-sample**" bucket on the left hand-side to unfold it and see what scope is under the bucket. Unfold the "inventory" scope to see what collections are under this scope. Scopes and collections segregate data of different types for more efficient indexing, querying and resource utilization.

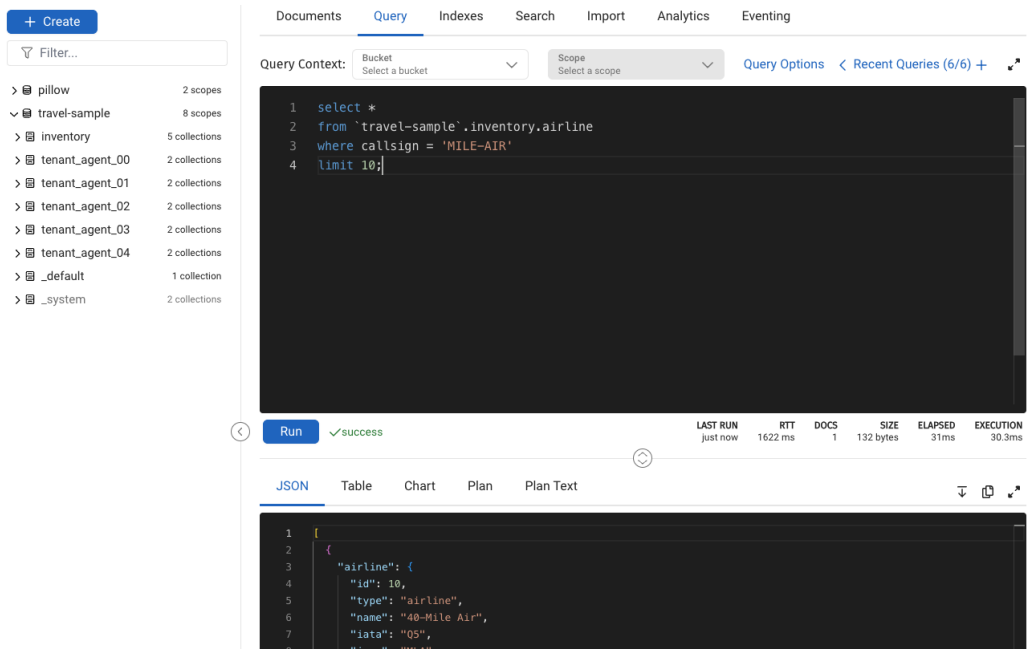


3. To do an initial data exploration, click on “**Documents**” and select the “**travel-sample**” bucket then a scope and a collection of your choice to review a subset of documents. You can click on the Doc ID of any document to see the content of that document.

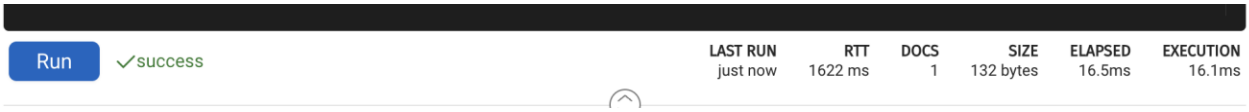


4. In the “**Query**” tab data can be queried using SQL++. If you know SQL, then you will find the syntax almost identical making querying of data very easy. For instance, to get a list of 10 airlines this query could be used:

```
SELECT *
FROM `travel-sample`.inventory.airline
WHERE callsign = 'MILE-AIR'
LIMIT 10;
```

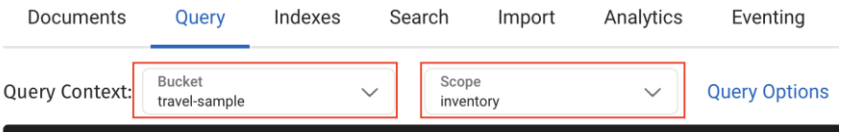


5. Note the statistics under the query editor window. The "**EXECUTION**" time represents the time it takes to the database to run the query. The "**RTT**" is the Round-Trip Time, which is the total time elapsed from the user perspective. It includes the execution time plus the transfer over the network, in other words the latency due to the distance between the client and the Couchbase database. To minimize this time, the Couchbase cluster should always reside in the same region as the client server



6. Explore the different views on the result set pane. Results may be displayed as JSON (native), tabular for convenience, and you can have a look at the execution plan to identify which index was used, and which step took the most time. "Chart" let you build visualization very easily.

7. To simplify the writing of SQL++ queries, it is possible to specify a default bucket and scope. This is known as setting the context of the query. In the UI, locate the context settings and position the default bucket to "**travel-sample**" as well as the default scope as "**inventory**".



8. Then you can run the same query with a simpler syntax:

```
SELECT *
FROM airline
WHERE callsign = 'MILE-AIR'
LIMIT 10;
```

9. Run another query, this one adds a filter clause:

```
SELECT name, address, city, description
FROM hotel
WHERE country= "United Kingdom";
```

10. This next query aggregates the number of reviews, stored in lists/arrays:

```
SELECT city, SUM(ARRAY_COUNT(reviews)) AS num_reviews
FROM hotel WHERE country="United Kingdom"
GROUP BY city
ORDER BY num_reviews DESC
```

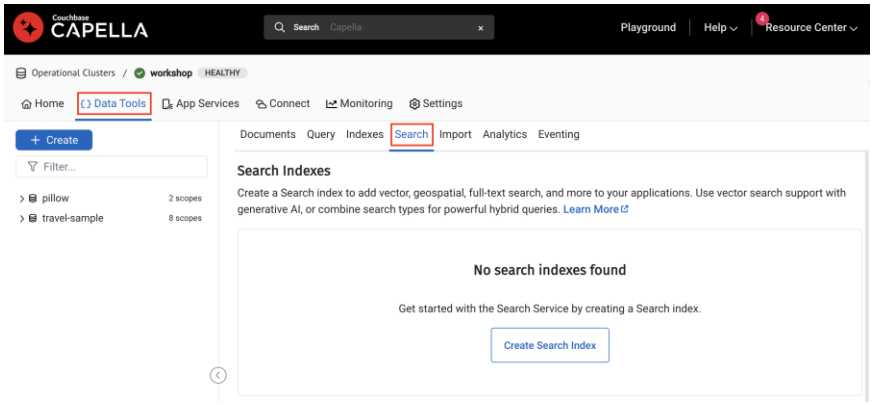
11. Feel free to make up your own queries to explore SQL++ capabilities

B. Full-text Search

Note: Before using full-text search, it is necessary to know the schema of the document, especially the access path of the fields you want to search. In this exercise, we will index the hotel collection. This is the schema of the documents. Highlighted are the fields that will be indexed:

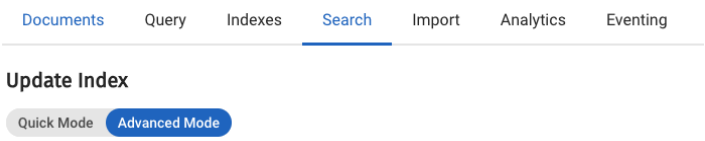
```
hotel                                                    917 documents
  address null | string
  alias null | string
  checkin null | string
  checkout null | string
  city null | string
  country string
  description string
  directions null | string
  email null | string
  fax null | string
  free_breakfast boolean
  free_internet boolean
  free_parking boolean
  geo object {
    accuracy string
    lat number
    lon number
  }
  id number
  name string
  pets_ok boolean
  phone null | string
  price null | string
  public_likes [string]
  reviews [object]
    author string
    content string
    date string
    ratings object {
      Business service number
      Business service (e.g., internet access) number
      Check in / front desk number
      Cleanliness number
      Location number
      Overall number
      Rooms number
      Service number
      Sleep Quality number
      Value number
    }
  state null | string
  title string
  tollfree null | string
  type string
  url null | string
  vacancy boolean
```

1. From the root screen, in the “**Operational**” menu, click on your cluster name to enter the database interface, and go to “**Data Tools**” menu, then select “**Search**” to access the Full-text Search workbench.

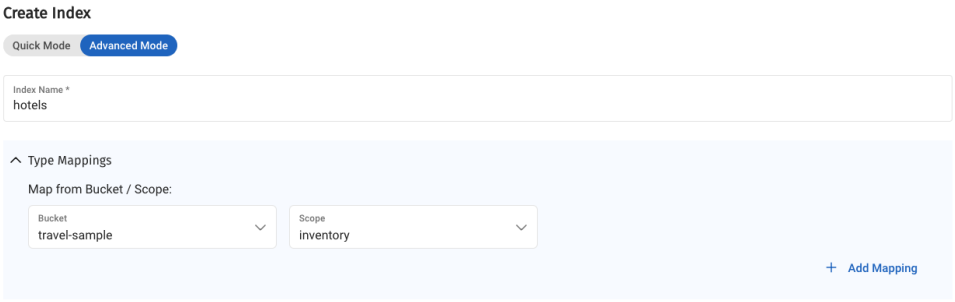


2. Click on the “**Create Search Index**” button to create your first full-text search index. You have 2 modes for creating an index: the “Quick Mode” and the “Advanced Mode”. “Quick mode” is more convenient for simple indexes, its wizard facilitates and speed up the process of index creation by inferring the schema and setting parameters automatically. In this example, as we will define complex structures (objects and nested objects), toggle

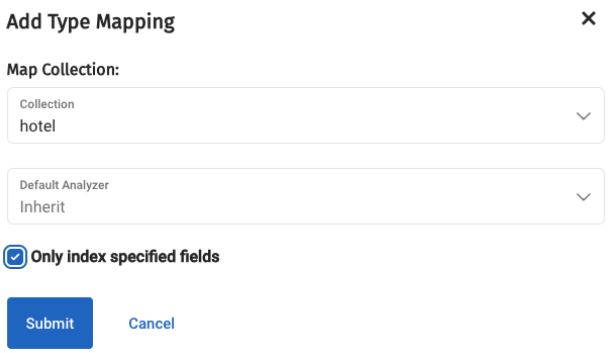
to “**Advanced Mode**”.



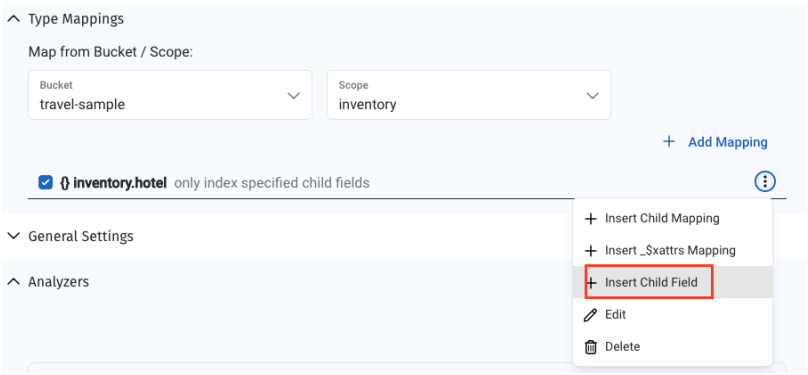
3. Enter “**hotels**” as the index name. Then choose the bucket “**travel-sample**” and the scope “**inventory**”.



4. Then click on “**Add Mapping**” and choose the “**hotel**” collection to search in. Leave the Default Analyzer as “**Inherit**” and check the box “**Only index specified fields**”.

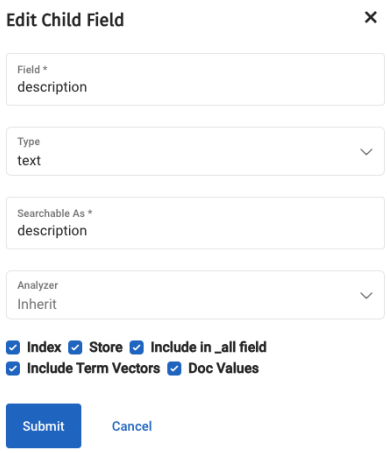


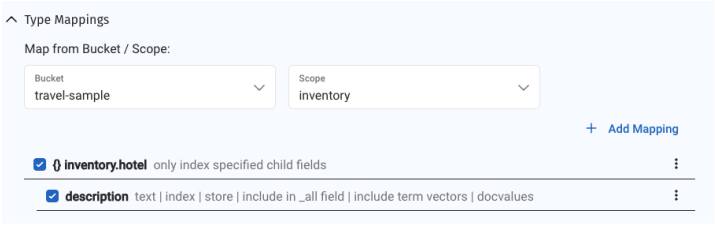
5. Click on the 3 dots near the “**inventory.hotel**” line and “**Insert Child Field**”



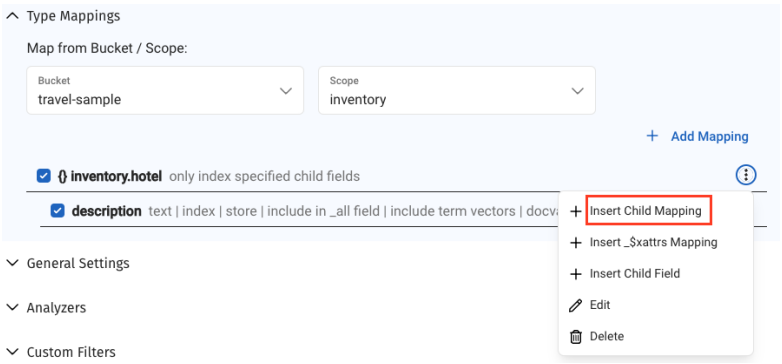
6. Name it “**description**” and check all the boxes. It determines which metadata will be included in the index, enabling different usage of the index. Then click “**Submit**”

   Attention: At the time of writing this lab, there might be a bug in this interface that let it display only one option “**Index**”. In that case, “**Submit**” the index, then edit it by clicking on the 3 dots near “description” then “**Edit**” and check all boxes that appear now.

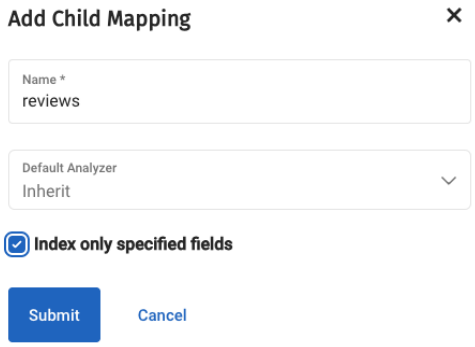




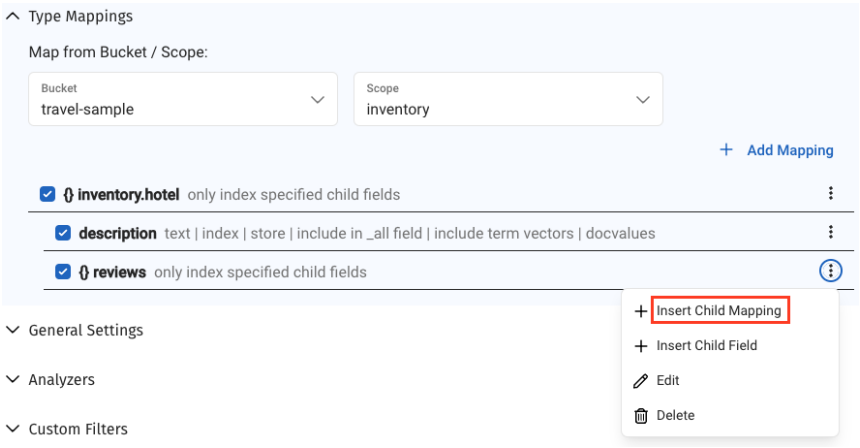
7. Then, the field “reviews” is an object. To declare it, click on the 3 dots near “**inventory.hotel**” and “**Insert Child Mapping**”



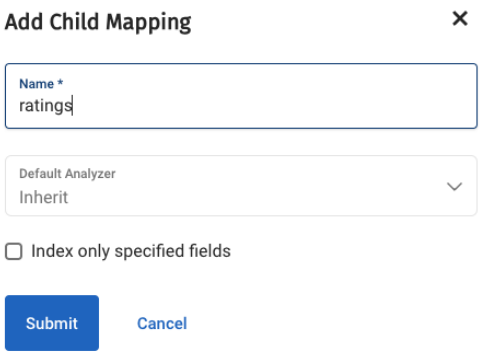
8. Name it “**reviews**” and check the box “**Only index specified fields**”



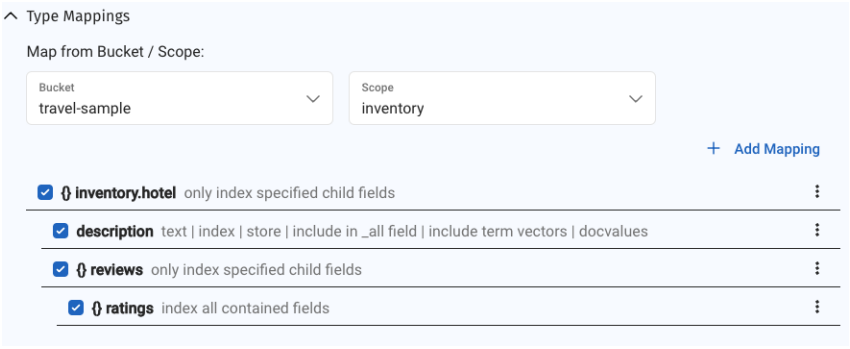
9. Again, “**ratings**” is an object within “**reviews**”, so click on the 3 dots near “**reviews**” and “**Insert Child Mapping**”.



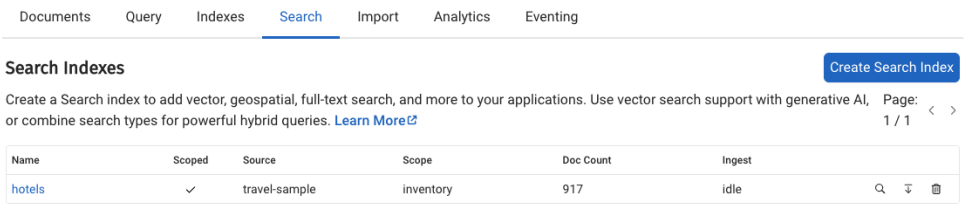
10. Name it “**ratings**” and leave the box “**Only index specified fields**” **<u>unchecked</u>** so that it will index every field under ratings. Then “**Submit**”.



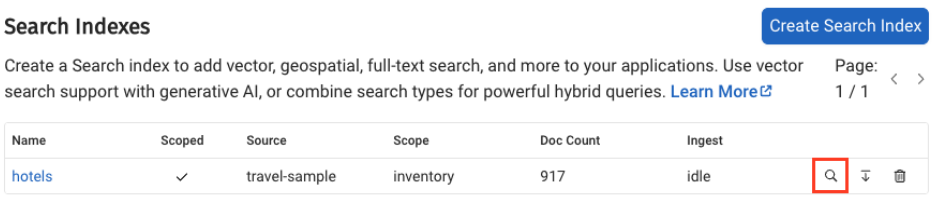
11. Your index mapping should look like this.



12. Finally, click on "**Create Index**" at the bottom to create the index. Couchbase will then start scanning and indexing the field. Once the index is complete, you will see 917 documents count in the index.



13. To test the index, click on the magnifier next to the index name



14. Try to search the following terms and observe the results returned:
    - description:location (see results derivated from the root *locat*)

- description:hotel
- description:hotel~1 (fuzzy search that allows 1 character difference in the results set)
- reviews.ratings.Cleanliness:5
- reviews.ratings.Cleanliness:>3

Congratulations, you have successfully defined indexes to run SQL++ queries as well as full-text searches. Those are the most popular ways to manipulate data stored in Couchbase. But there are other ways to interact with your data. While “Eventing” and “Analytics” are not available with a trial account, these services will be demonstrated by the instructor.

END OF WORKSHOP INSTRUCTIONS DAY 1