# **Langflow installation**

Langflow is a low-code app builder for RAG and multi-agent AI applications. It’s Python-based and agnostic to any model, API, or database.

# Cloud: DataStax Langflow is a hosted environment with zero setup. Login to Astra account which would navigate into Langflow directly.

# Self-managed: Run Langflow in your environment.Use theQuickstartguide below to install Langflow and to create and execute a flow.

## **Quickstart**

# You can install Langflow from pip: pip install langflow

# Next, run: python -m langflow or langflow run

# 

# **Langfuse installation**

# **Self-Hosting with Langfuse**

### **Localhost (docker)**

# Clone repository

git clone https://github.com/langfuse/langfuse.git

cd langfuse

# Run server and database

docker compose up -d

Langfuse is now running on your local machine.

* The server is accessible at: http://localhost:3000

Langfuse requires a persistent Postgres database to store its state. Docker compose has a database setup as well. So when Langfuse captures all metrics from the Langflow application, all trace data will be stored in the PostgreSQL database.

# 

# **Getting started with Astra Streaming**

Astra Streaming is a cloud native data streaming and event stream processing service tightly integrated into the Astra UI and powered by Apache Pulsar. Using Astra Streaming, create Pulsar instances, manage their clusters, scale across cloud regions, and manage Pulsar resources such as topics, connectors, functions and subscriptions.

## **Prerequisites**

You will need the following to complete this guide:

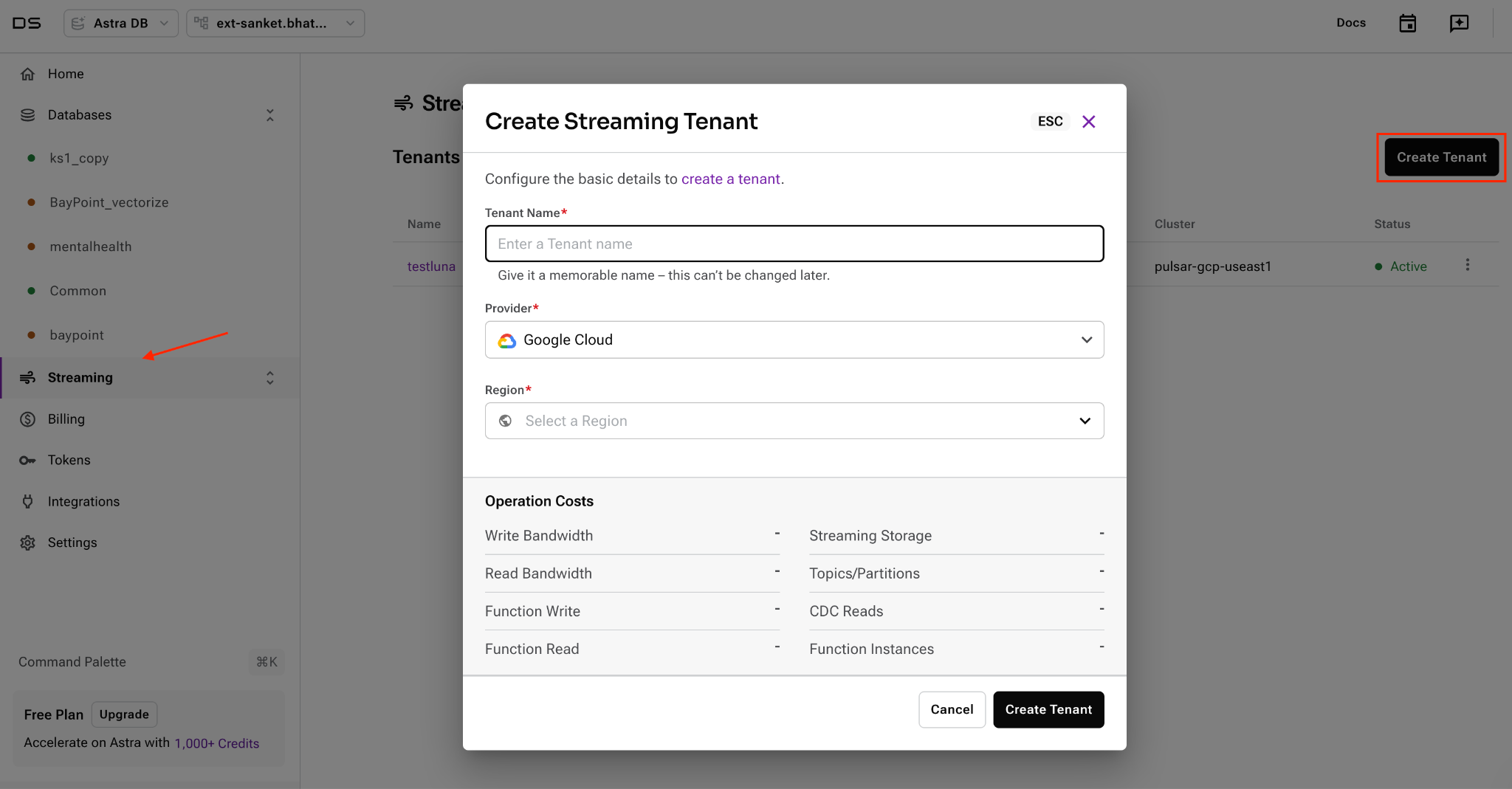
* Astra account

## **Streaming tenant**

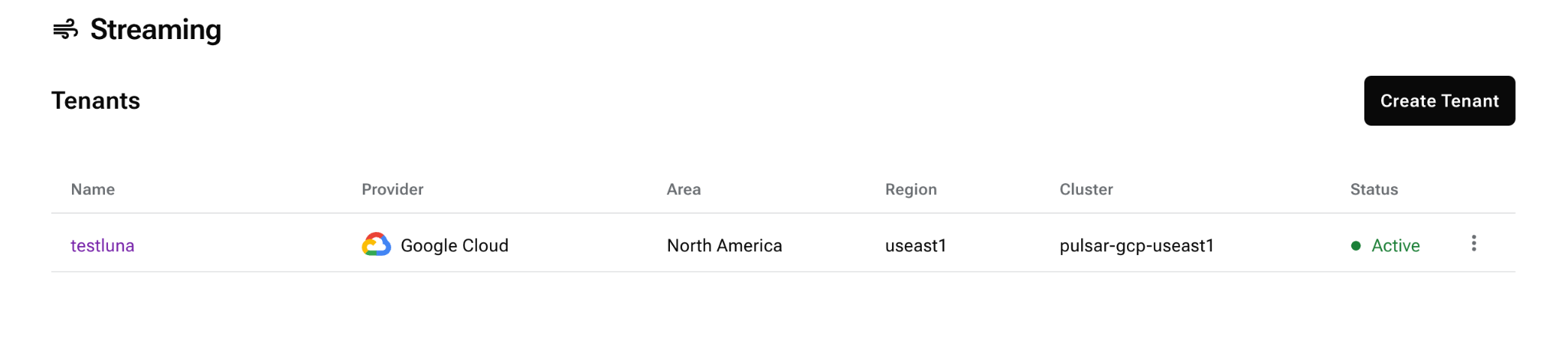
Tenant is a portion of DataStax’s managed Apache Pulsar. The steps in the below tabs will guide you through creating a Streaming Tenant. You’ll use this tenant to create namespaces, topics and functions The only difference between the tabs is how your tenant is created - they all have the same result.

*Astra Portal*

1.Once signed in, click the "Create Streaming " button on the portal home page.

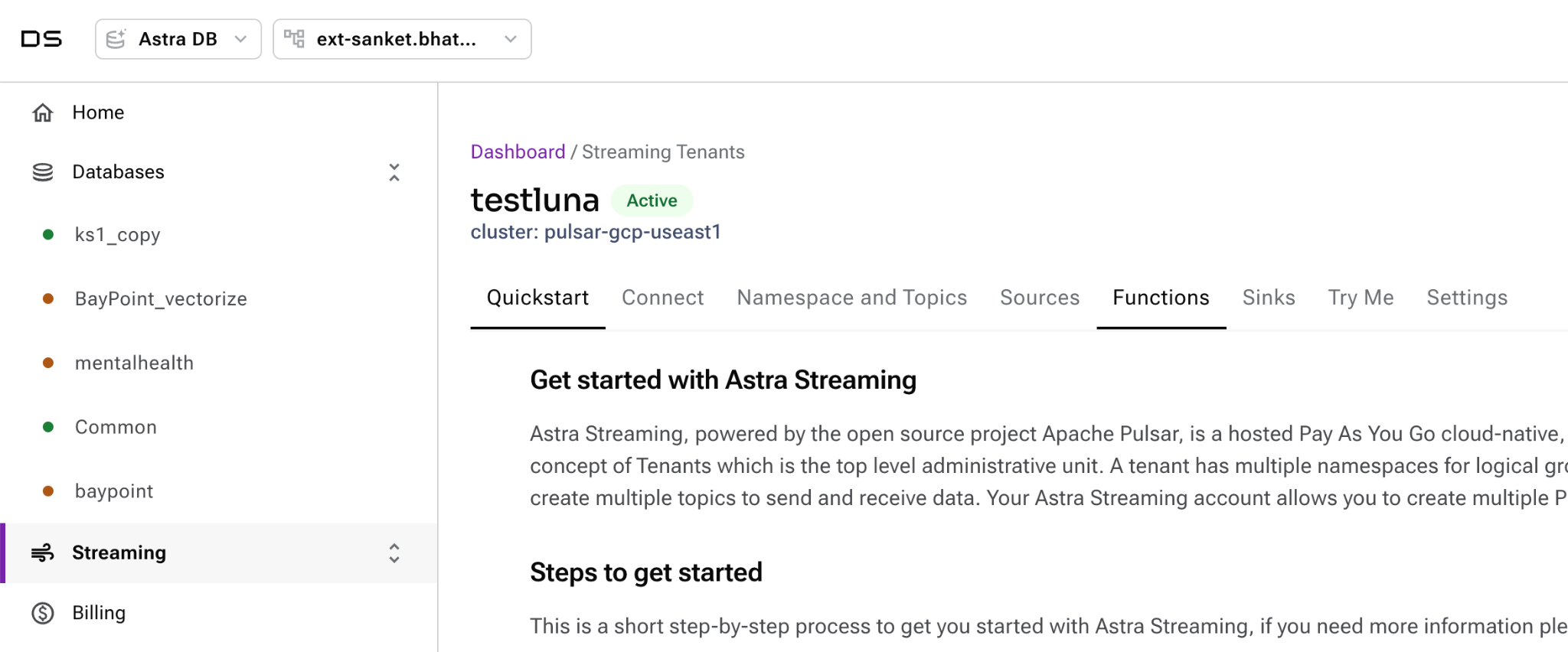


2. Name your Streaming Tenant.Choose your preferred cloud provider and region.



3. Click the "Create Tenant" button.

4. You will be directed to the quickstart page for your new tenant.

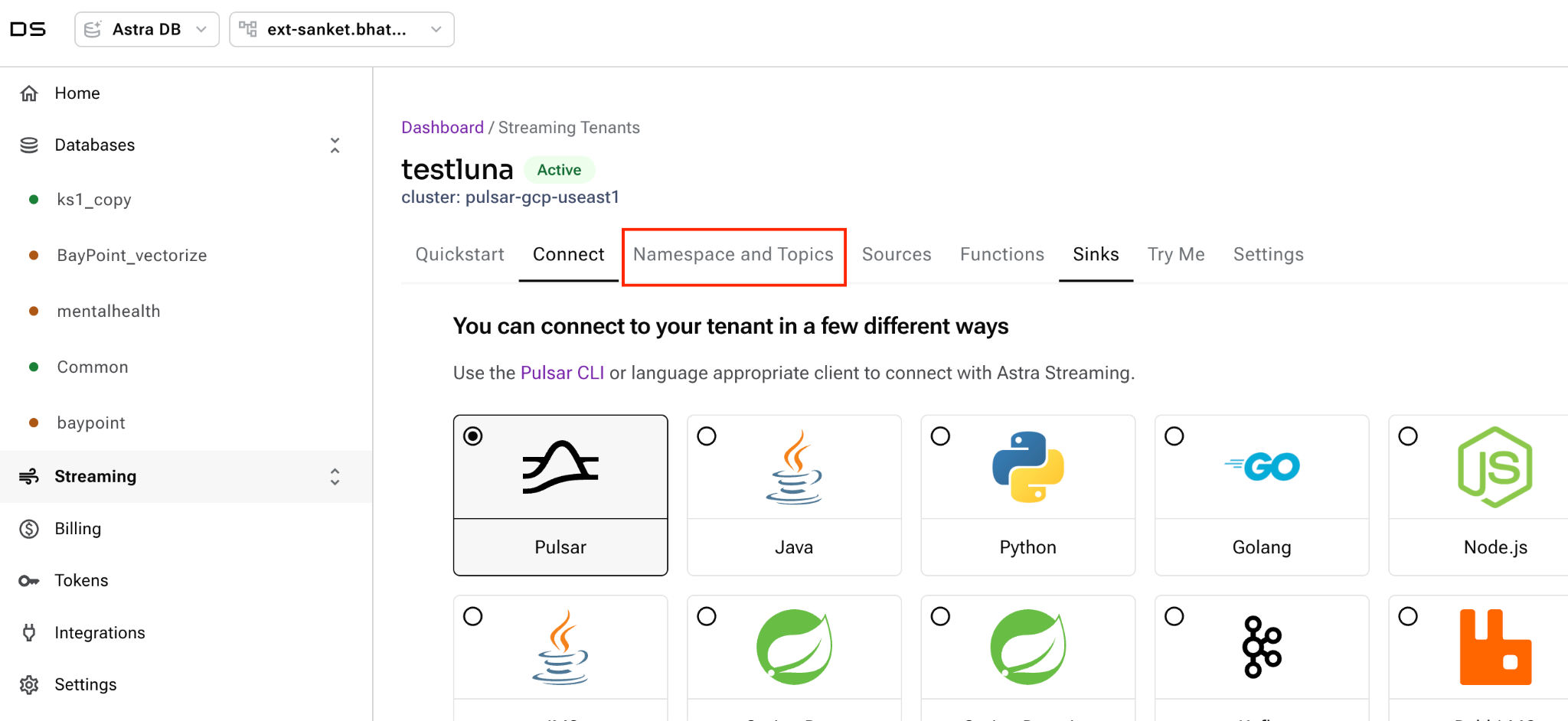


## **A namespace to hold topics**

A namespace exists within a tenant. A namespace is a logical grouping of message topics. Tenants usually have many namespaces.

*Astra Portal*

1. Navigate to the "Namespace And Topics" tab.



2. Click the "Create Namespace" button, and give your namespace name like "my-namespace".

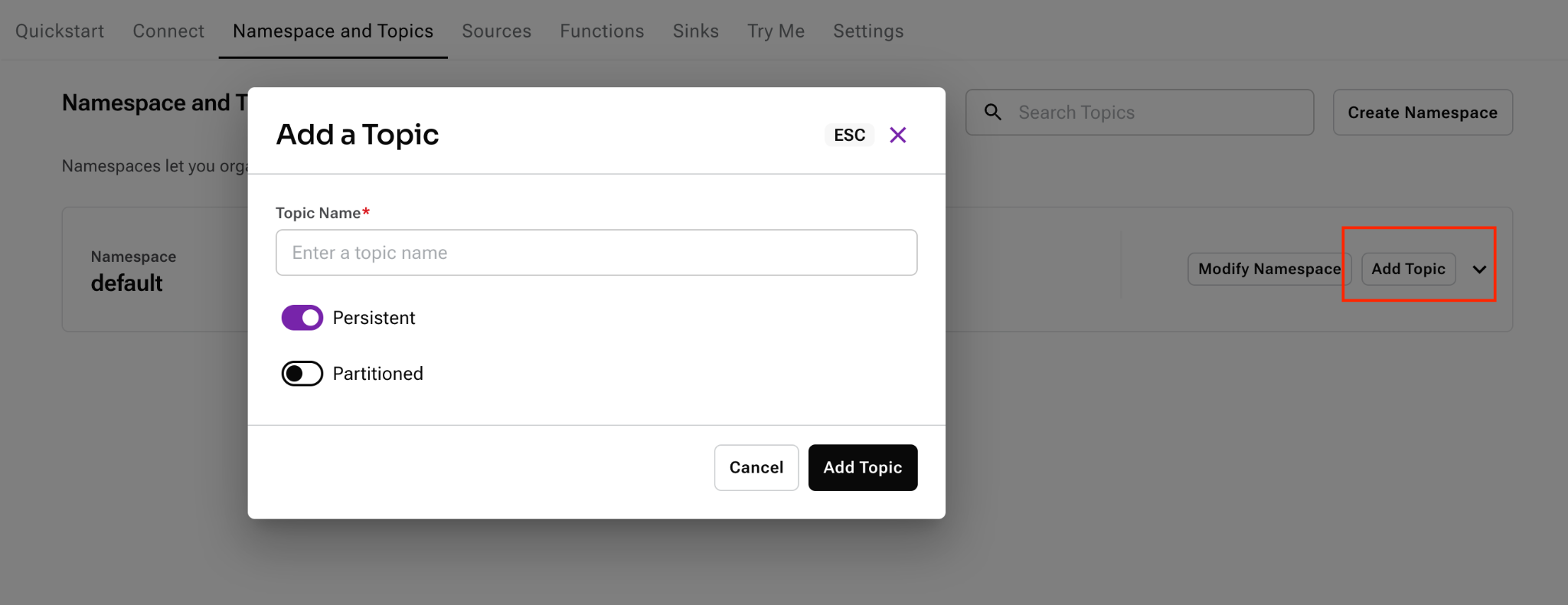
Note: Astra Streaming automatically created a namespace named "default" when you created your tenant. If you would like to use the default namespace instead of creating a new namespace.

## **A topic to organize messages**

Topics are the core construct of a messaging system. Topics provide a way to group messages matching certain criteria.Topics are where others can "listen" for new messages. Consumers subscribe to topics to "listen" for messages, and functions and connectors can also "listen" for messages and automate workflows. In Pulsar, topic addresses look like a URL (ie: persistent://tenant/namespace/topic)

*Astra Portal*

1. In the Namespace And Topics tab, locate the namespace created above and click its Add Topic button.
2. Provide a name for the topic.
3. Leave the choice of persistence and partitioning

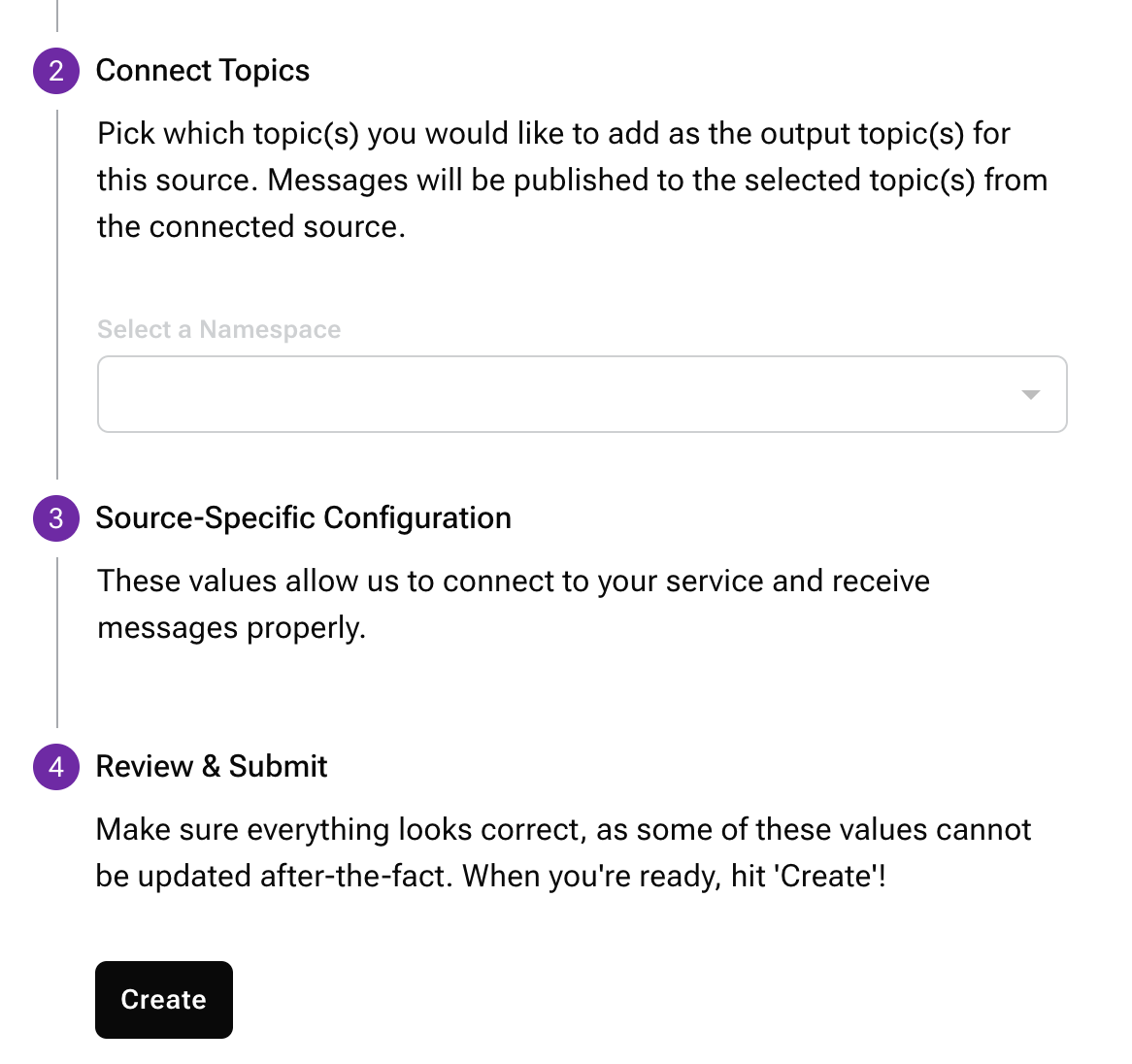
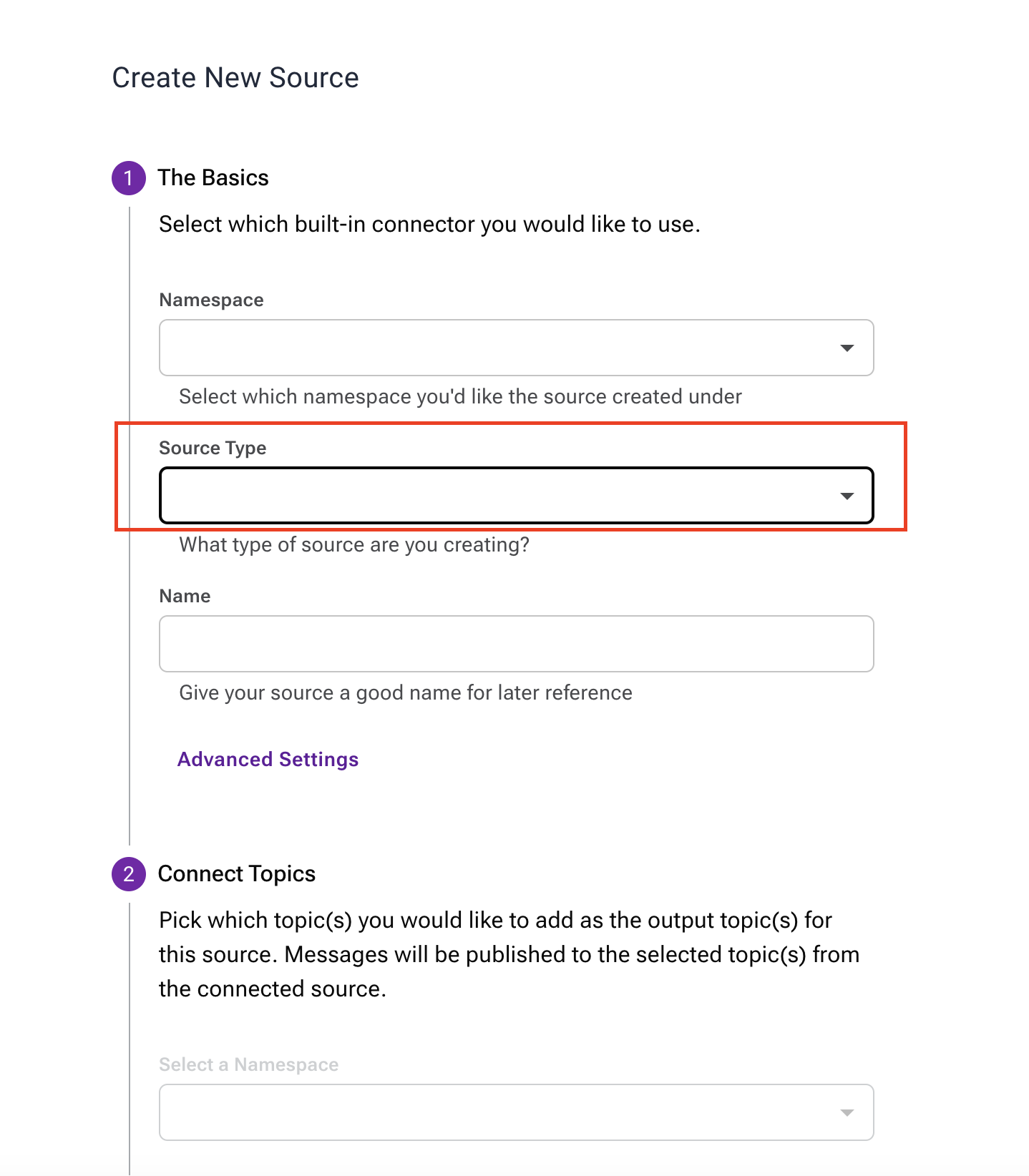


## **Sources**

Sources feed data from external systems such as other messaging systems, change data capture streams into Pulsar. Depending on what source you want to connect, enter the details of the source. In this demo, we have used a PostgreSQL database.

*Astra Portal*

1. In the Sources tab, create a new source and provide necessary details of the database as shown below.
2. Select source type from where the data will be sourced. For postgres database, type would be chosen as Debezium PostgreSQL
3. Select Topic which was created in the previous section.

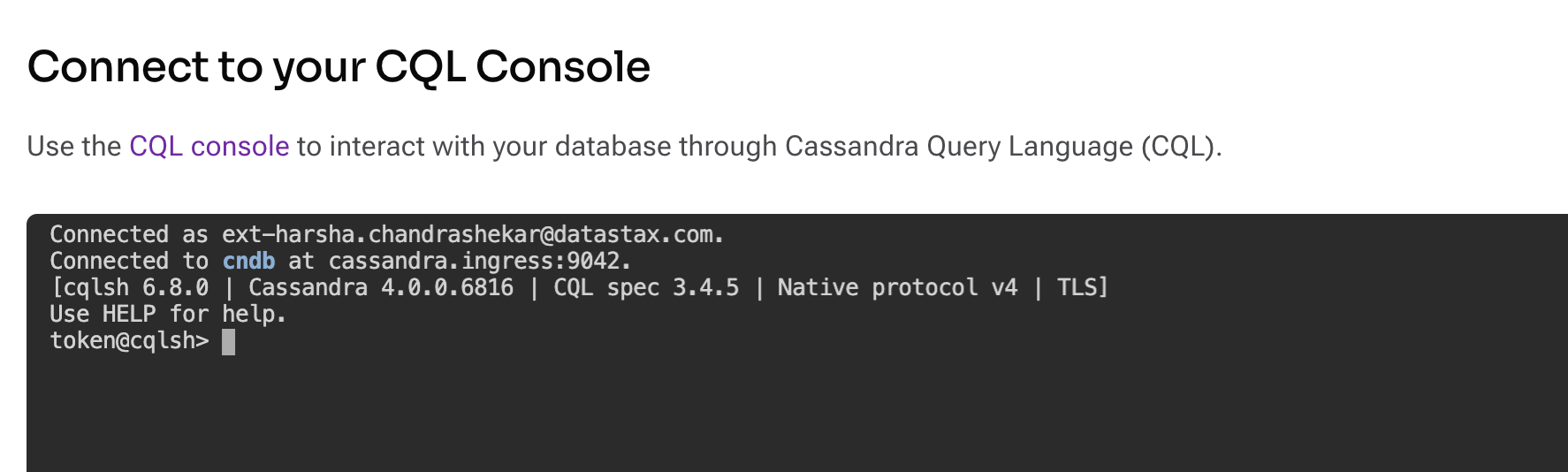


## **Astra DB - Table creation for streaming**

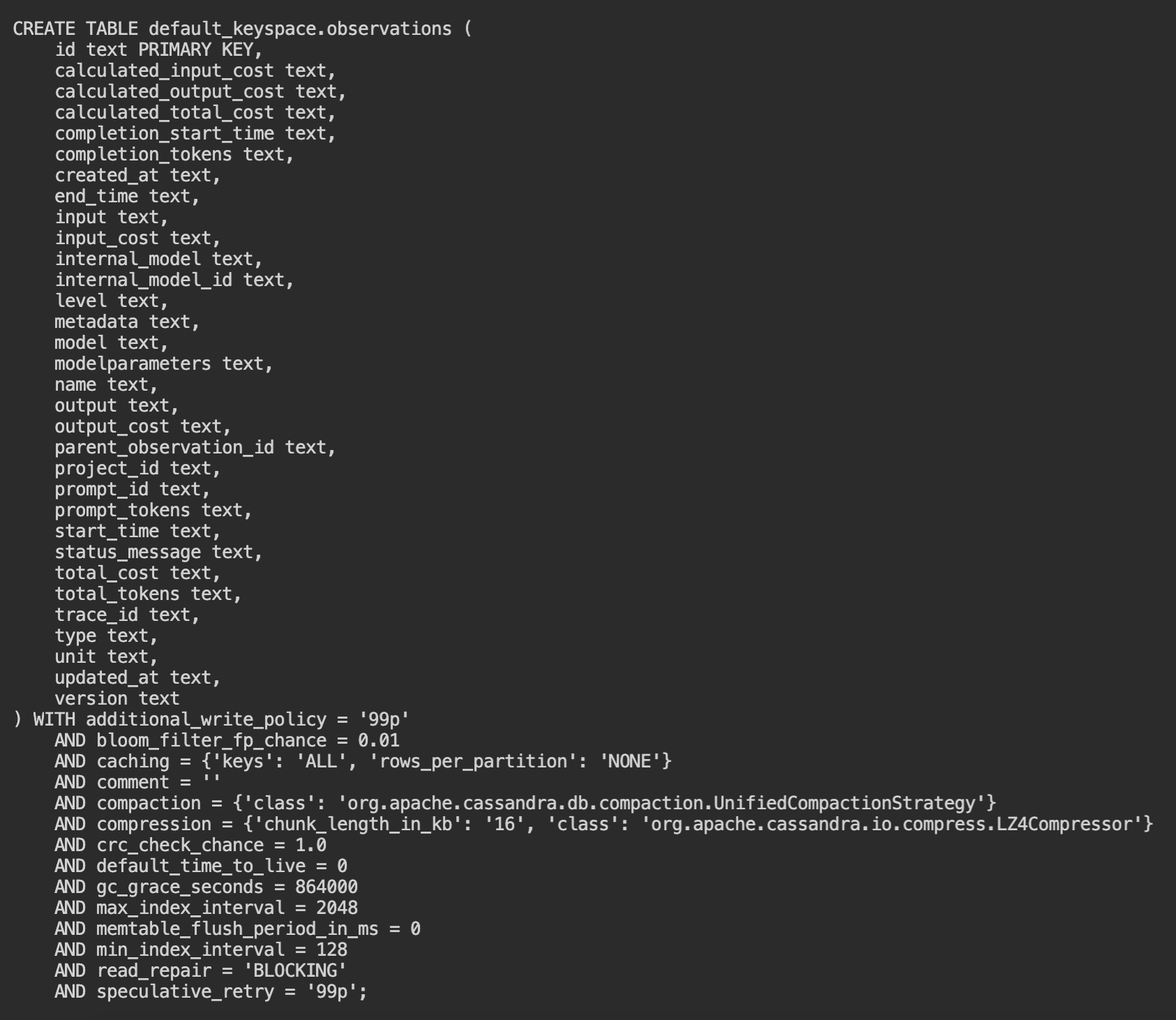
Navigate to the database account dashboard and select the database where tables have to be created.

*Astra Portal*

1. Select CQL Console which will open command line



1. Select the namespace where the table has to be created. Default will be created under default\_keyspace.
2. Create necessary tables required for streaming.

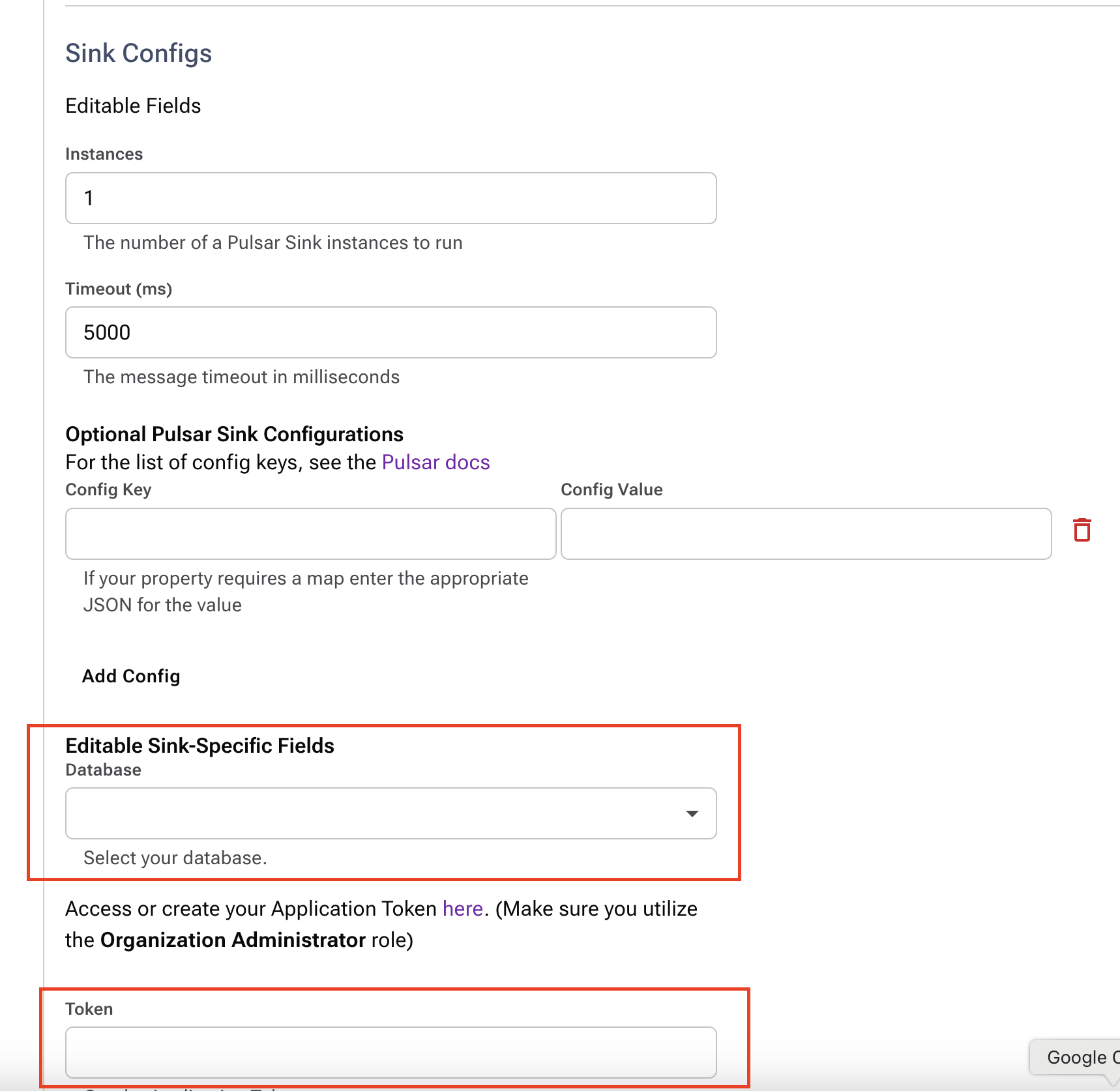
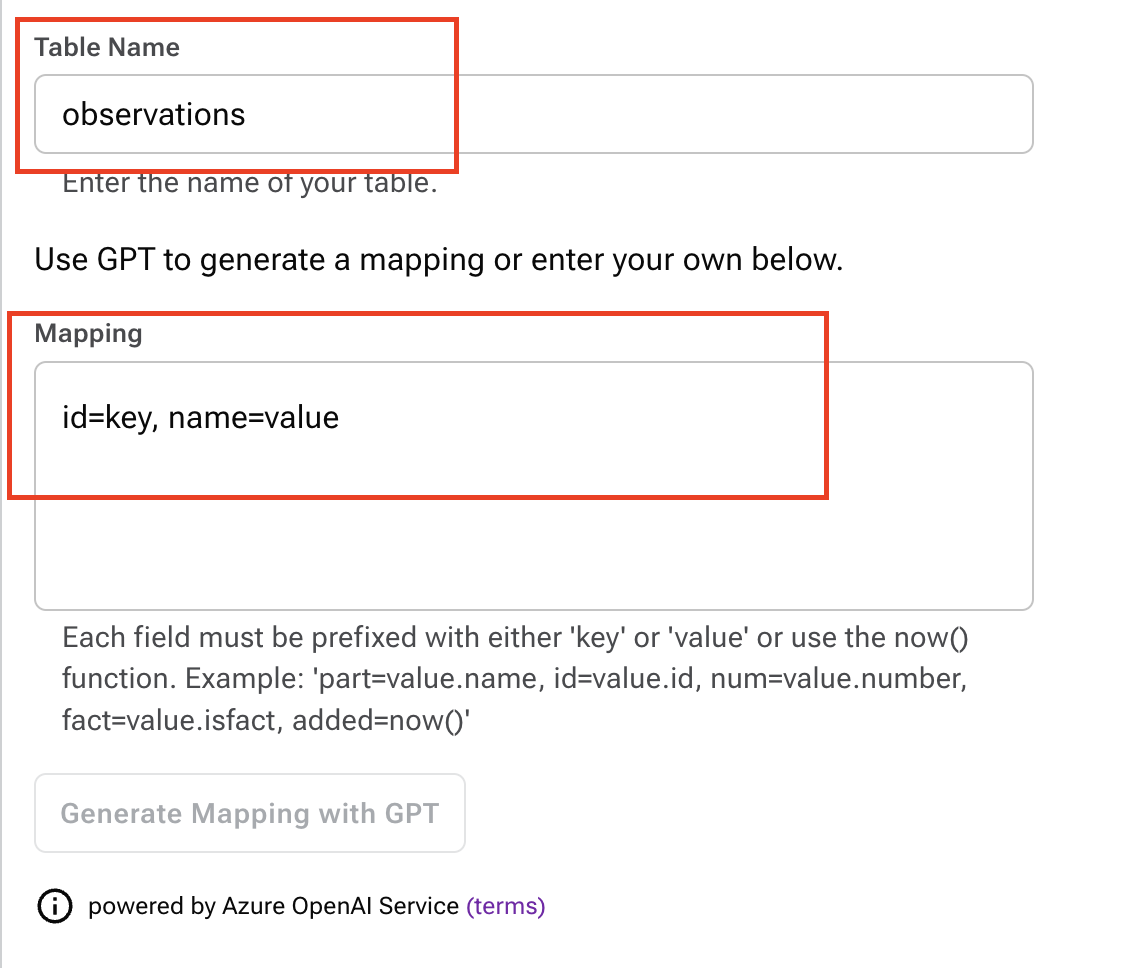


## **Sink connectors**

Sinks feed data from Pulsar topics into Astra DB

*Astra Portal*

1. In the Sinks tab, click Create Sink button
2. Complete the below highlighted details :

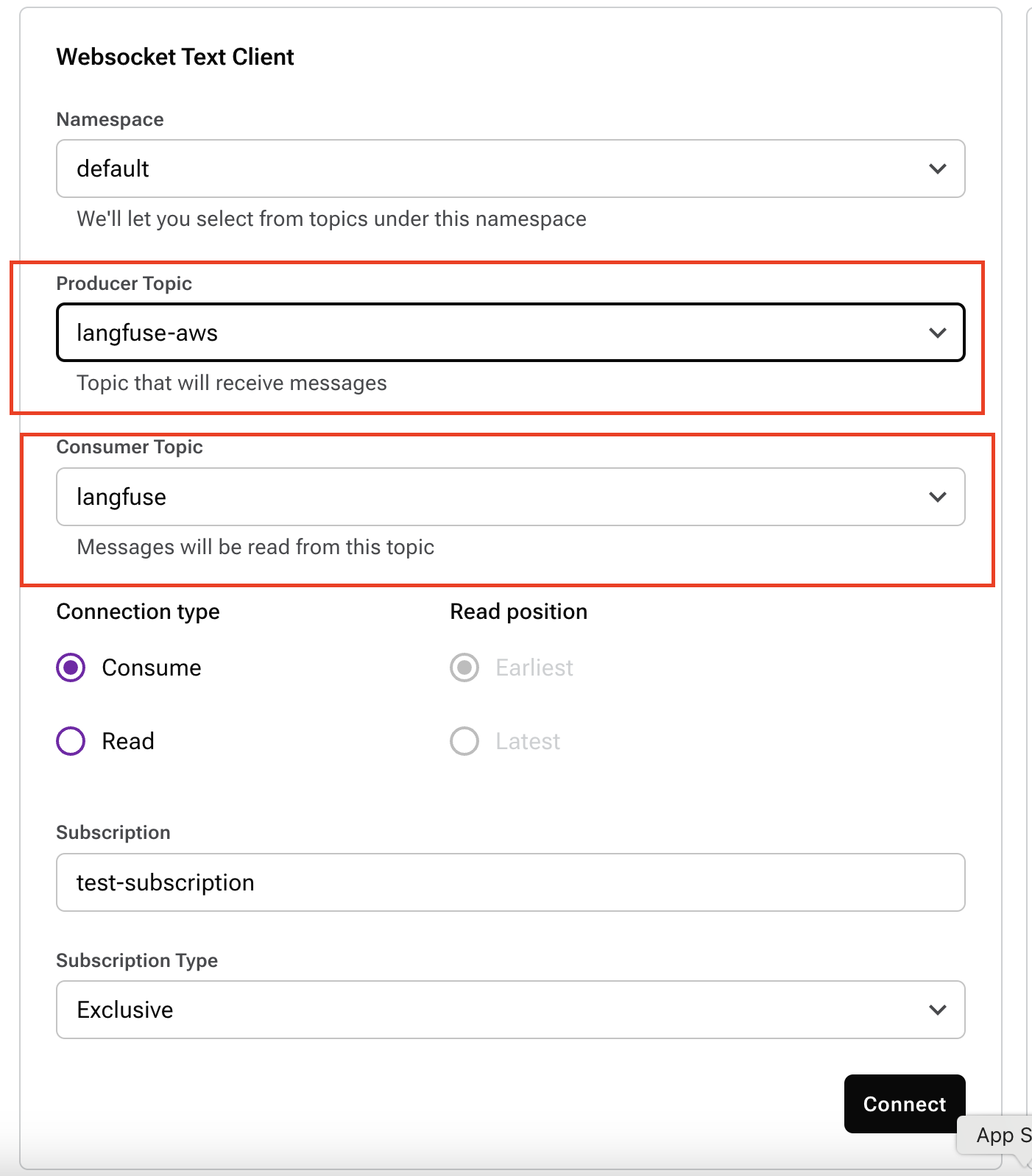
Note : Mapping has to be done in the template shown in the screenshot above.

1. Once the sink is created , it will start running and records will be loaded into Astra DB. Logs can be viewed on the same page if there are any errors.

## **Testing the Streaming**

*Astra Portal*

1. In the Try Me tab, select topics created before.



1. Once you click Connect , the Test message tab will be shown with the latest records.