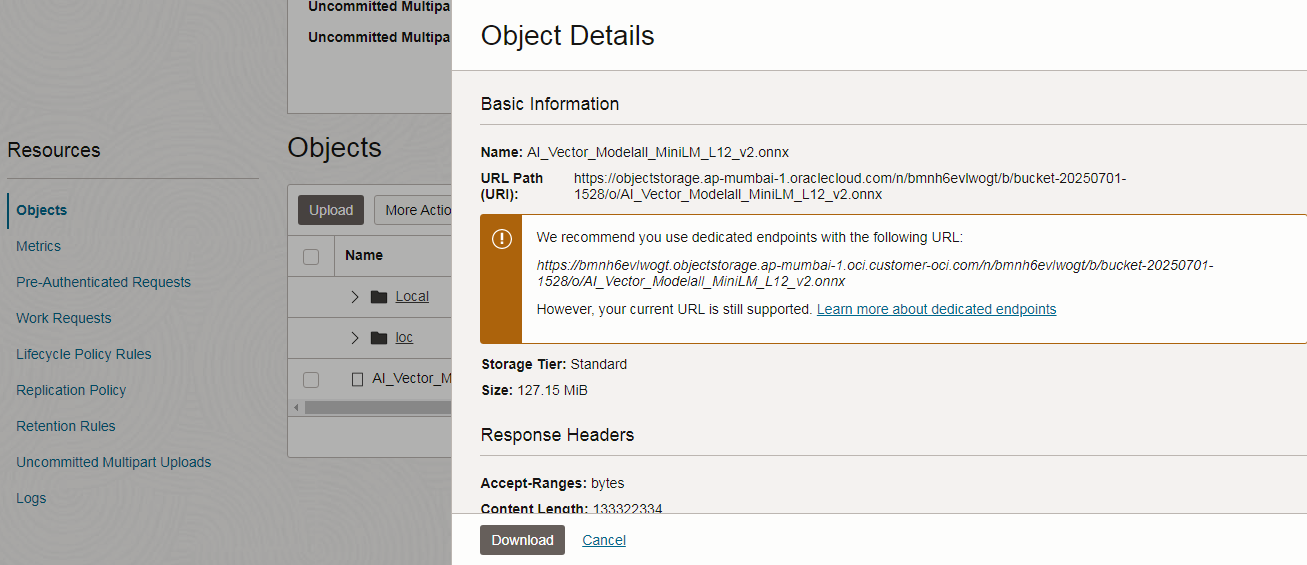
**AI Vector Search**

If you are going to use Vector Provider type as **“Database ONNX Model”** then firstly load the ONNX model into the database.

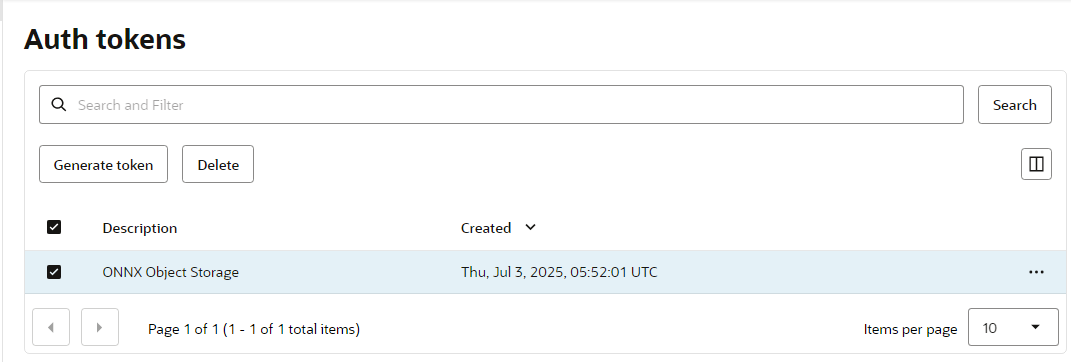
**Load ONNX Model into the Database**

To load the ONNX model into the database, we first upload the file to object storage and create the necessary credentials to access this file from the Autonomous Database. Let's get to it:

1. Upload the ONNX file to object storage. You can download the pre-built ONNX model using this [link](https://adwc4pm.objectstorage.us-ashburn-1.oci.customer-oci.com/p/VBRD9P8ZFWkKvnfhrWxkpPe8K03-JIoM5h_8EJyJcpE80c108fuUjg7R5L5O7mMZ/n/adwc4pm/b/OML-Resources/o/all_MiniLM_L12_v2_augmented.zip).



1. After uploading the ONNX file to object storage, we have to create an Auth Token to access the file uploaded to object storage above.



1. Next, we grant necessary privileges to the parsing schema as user ADMIN. The user needs to connect to the database as ADMIN using Database Actions.  
     
   We grant the privilege to execute dbms\_cloud to the workspace parsing schema logged in as ADMIN. This allows the parsing schema to create credentials to access the ONNX file stored in the Object Storage while loading it into the database.

**GRANT execute on dbms\_cloud To <Your Schema Name>;**

**GRANT create mining model TO <Your Schema Name>;**

1. We start by creating a credential based on the Auth Token created above to access the file in object storage from the Autonomous Database. This ensures that the Oracle Database can securely access and interact with the file uploaded to object storage.

This PL/SQL snippet creates a credential named onnx\_obj\_store\_cred using the dbms\_cloud.create\_credential procedure. The credentials (username and AUTH Token) are used for authentication with OCI Object Storage.

**BEGIN**

**dbms\_cloud.create\_credential (**

**credential\_name => 'onnx\_obj\_store\_cred',**

**username => <Your Username>',**

**password => '<Auth Token>'**

**);**

**END;**

1. We load ONNX embedding model into the database using DBMS\_VECTOR.LOAD\_ONNX\_MODEL:

**BEGIN**

**DBMS\_VECTOR.LOAD\_ONNX\_MODEL(**

**model\_name => 'doc\_model',**

**model\_data => dbms\_cloud.get\_object (**

**credential\_name => 'onnx\_obj\_store\_cred',**

**object\_uri =>’ Your Object Storage URI>’, -- blob**

**metadata => JSON('{**

**"function" : "embedding",**

**"embeddingOutput" : "embedding" ,**

**"input":{"input": ["DATA"]}**

**}')**

**);**

END;

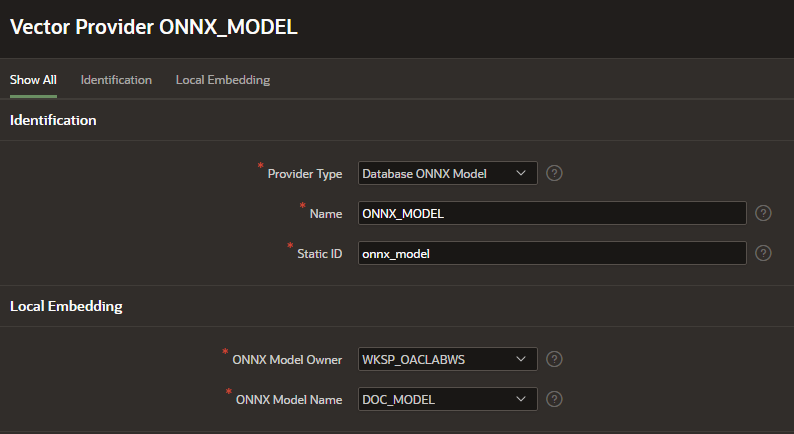
**Create a Vector Provider**

1. Navigate to Workspace Utilities via App Builder on the Oracle APEX Workspace home page. Then, click on Vector Providers, which is introduced with Oracle APEX 24.2.
2. Click **Create** to start and then enter a unique name and **Static ID** for the Vector Provider. The Static ID must be unique and will be referenced when using the service with the APEX\_AI package, APEX\_AI.GET\_VECTOR\_EMBEDDINGS.
3. Select a Provider Type:

* **Database ONNX Model**:
* Specify ONNX Model Owner (choose from schemas assigned to your workspace).
* Select the ONNX Model Name (the model to generate embeddings).
* **Generative AI Service**: Select AI Provider. Options include OCI, OpenAI, or Cohere. APEX will automatically adjust the interface upon selection.
  + **Custom PL/SQL**: Enter the Custom Function Name (a PL/SQL function that converts input into embeddings and returns a VECTOR data type).

Click **Create** to save.

1. You can choose ‘ONNX Model Name’ to DOC\_MODEL.



1. Execute below query to add Vector column in **eba\_project\_milestones** table**.**

**ALTER TABLE eba\_project\_milestones ADD (vector\_column vector);**

Update the vector\_column to store embeddings for name and description columns. (p\_service\_static\_id is vectore provider static ID created in previous step).

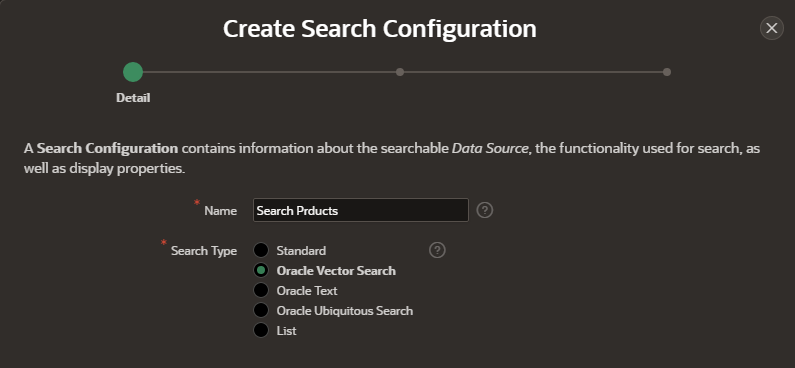
**UPDATE eba\_project\_milestones**

**SET vector\_column = apex\_ai.get\_vector\_embeddings(**

**p\_value => name || ' ' || description,**

**p\_service\_static\_id => 'onnx\_model' );**

1. Navigate to the Workspace home page, click App Builder, and select your application. Open Shared Components and click Search Configurations. Click Create. Name your configuration (e.g., Search Milestones - Vector), choose Oracle Vector Search as the type, and click Next.



1. Set up the source by selecting a Vector Provider to generate embeddings (e.g., ONNX Model). Pick the source type (Table or SQL Query) and choose the table owner and table name containing the vector column (e.g., POPULAR\_MOVIES). Click Next.

Map the columns. To create a Search Type as Oracle Vector Search, we need to map the Vector Attributes along with Column Mapping.

**Primary Key Column:** Select the key column (e.g., ID).

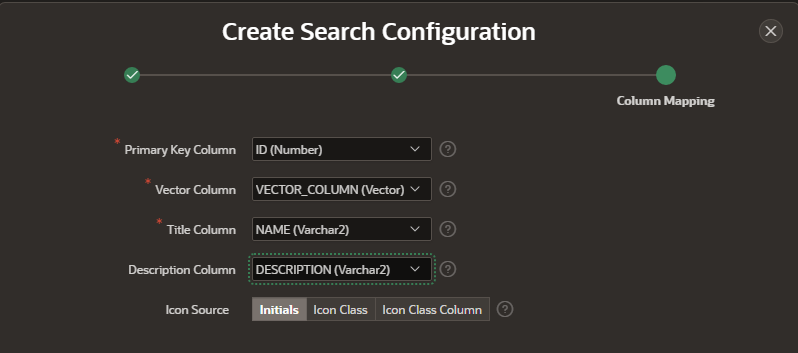
**Vector Column:** Select the vector data column (e.g., VECTOR\_COLUMN).

**Title Column:** Choose the title column (e.g., NAME).

**Description Column:** Pick the description column (e.g., DESCRIPTION).

**Icon Source:** Set how icons appear, such as using initials.

Click Create Search Configuration to save it.



1. Creating an Oracle Vector Search Page

The Create Page Wizard helps you set up a search feature effortlessly. It creates a search field and allows you to select search configurations during setup. You can also add more search configurations later in Page Designer if needed.

Check out this [link](https://docs.oracle.com/en/database/oracle/apex/24.2/htmdb/creating-an-oracle-vector-search.html#GUID-C0BEB835-CC6F-4499-B8EB-0334AE646F0C) to learn more about how to create a Search Page in Oracle APEX.

1. Run the app and perform different searches to observe output.

