```
# Step 1: Install necessary packages (run in a notebook cell)
!pip install -U chromadb langchain-chroma langchain_openai
```

```
# Step 2: Import libraries with updated imports
from langchain_chroma import Chroma
from langchain_openai import OpenAIEmbeddings
from langchain_community.document_loaders import TextLoader
from langchain_text_splitters import RecursiveCharacterTextSplitter
import os

# Step 3: Set your OpenAI API key (replace with your key)
os.environ["OPENAI_API_KEY"] = "Your-api-key"

embeddings = OpenAIEmbeddings()
```

```
# Step 4: Load your documents (replace 'your_file.txt' with your file path)
loader = TextLoader('notes.txt')
documents = loader.load()
```

```
# Step 5: Split the documents into chunks for better embeddings
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000,
chunk_overlap=200)
chunks = text_splitter.split_documents(documents)
```

```
# Step 6: Initialize the OpenAI embeddings
embeddings = OpenAIEmbeddings()
```

```
# Step 7: Create and persist the Chroma vector store
persist_directory = './chroma_langchain_db'
vectorstore = Chroma.from_documents(
    documents=chunks,
    embedding=embeddings,
    collection_name='example_collection',
    persist_directory=persist_directory
)
```

```
# Step 8: Load the persisted vector store (can be in a new notebook cell)
vectorstore = Chroma(
    collection_name='example_collection',
    embedding_function=embeddings,
    persist_directory=persist_directory
)
```

```
# Load the persisted Chroma vector store (replace with your actual variables
if different)
loaded_vectorstore = Chroma(
    collection_name='example_collection',        # Your chosen collection name
    embedding_function=embeddings,               # The embeddings function
instance you set previously
    persist_directory='./chroma_langchain_db'    # Path used for persistence
earlier
)
```

```
# Step 9: Perform similarity search on the loaded vector store
query = "what is benchmark"
results = loaded_vectorstore.similarity_search(query, k=3)
for doc in results:
    print(doc.page_content)
```