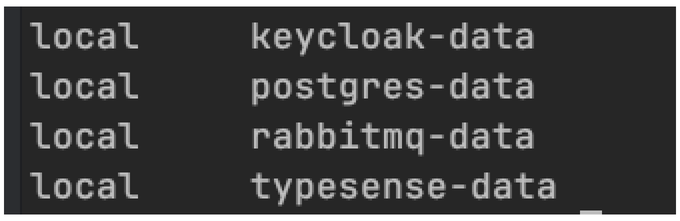
Answers Challenge

Preparation

1. You may want to clean the DBs before beginning this so that you have clean data in the DB. To do this we can delete the volumes in docker and then restart the AppHost. Run the following command to check the persistent volumes in docker:

```
docker volume ls
```

2. You should see the following:



```
local     keycloak-data
local     postgres-data
local     rabbitmq-data
local     typesense-data _
```

3. To delete the postgres-data and the typesense-data run the following:

```
docker volume rm typesense-data postgres-data
```

4. Now restart the AppHost and you will have clean databases. Create a few questions using postman so you have something in both DBs

Question Service

1. Create a new entity called "Answer.cs" using the info from the specification. Ensure you configure this as a "fully defined relationship" with the Question entity.
2. Add another property to the Question.cs to track the AnswerCount as per the specification.
3. Add the Answer as a new DbSet in the DbContext so we can query the answers table directly.
4. Create a new migration for the QuestionDbContext. Check the migration and ensure that:
    1. The QuestionId column in the Answers table is not nullable
    2. The onDelete behavior in the constraints is ReferentialAction.Cascade
5. Create a CreateAnswerDto.cs as per the specification.
6. Create the remaining API endpoints as defined below in the QuestionsController.

Method	Route	Purpose	Auth?	Returns
POST	/questions/{questionId}/answers	Create answer for question.	Auth	201 with **Answer**
PUT	/questions/{questionId}/answers/{answerId}	Update answer	Auth	204 no content
DELETE	/questions/{questionId}/answers/{answerId}	Delete answer. Do not allow accepted answers to be deleted.	Auth	204 no content
POST	/questions/{questionId}/answers/{answerId}/accept	Set answer as accepted answer. Ensure that the question does not already have an accepted answer.	Auth	204 no content

7. Use eager loading on the GetQuestionById endpoint to return the Answers along with the question.
8. Create the contracts for the events as defined in the specification.
    1. UpdatedAnswerCount
    2. AnswerAccepted

9. Publish the events from the respective endpoints. Note: **Typesense does not support atomic field updates (like increment or decrement)** so we have to publish the updated AnswerCount of the questions to ensure accuracy here.
    1. Post answer - AnswerCountUpdated
    2. Delete answer - AnswerCountUpdated
    3. Accept answer - AnswerAccepted

Search Service

1. Add the AnswerCountHandler.
2. Add the AcceptAnswerHandler
3. Restart the AppHost and test the new functionality in Postman.