RAG를 위한 Knowledge Base 설정

KB (Knowledge Base) for Amazon Bedrock 설정 방법 가이드 입니다.
이제 Bedrock 콘솔에서 RAG 작업을 쉽게 진행하세요!

19 July 2024

김제삼 (Jesam Kim)

Solutions Architect
AWS

1. KB 생성

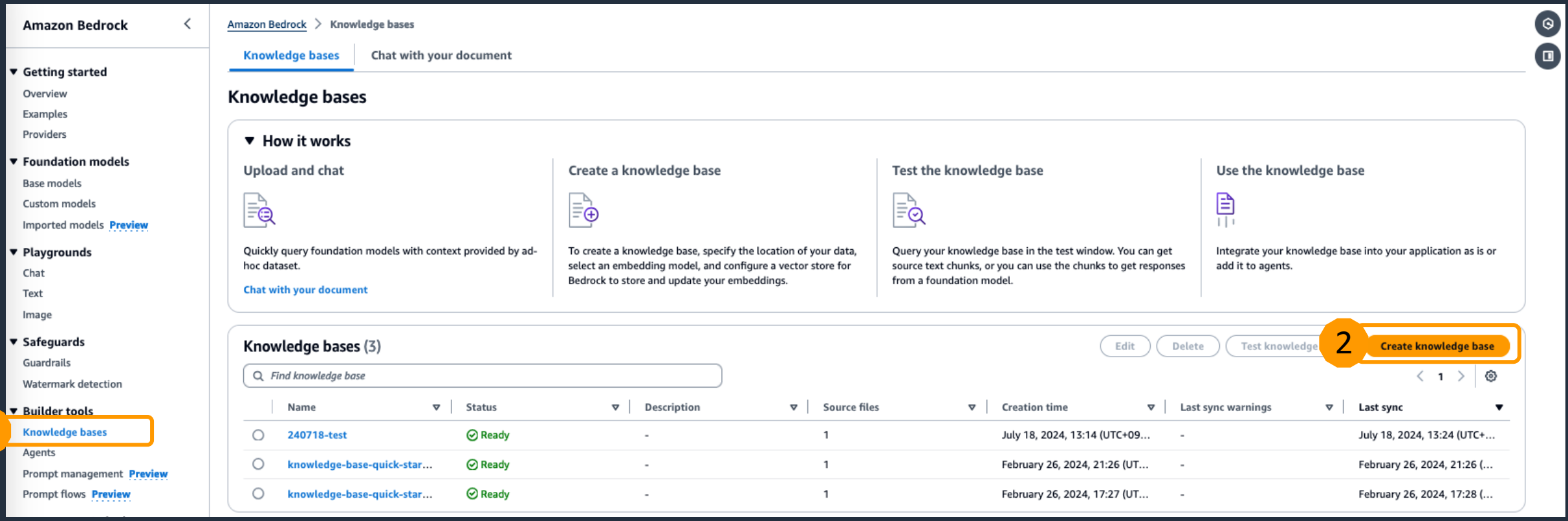
Amazon Bedrock 콘솔에 접속 합니다.

(1) Knowledge bases 메뉴에 접속 합니다.

(2) Create knowledge base 를 클릭하여 새로운 KB를 생성 합니다.

** KB를 만드는 것은 OpenSearch Serverless 컬렉션을 하나 만드는 것 입니다. 여기서 인덱스 구성 정보(청킹 등) 설정이 표함 됩니다.*

즉, KB를 만드는 것은 하나의 Index를 만드는 것과 같습니다.



2. KB 세부사항

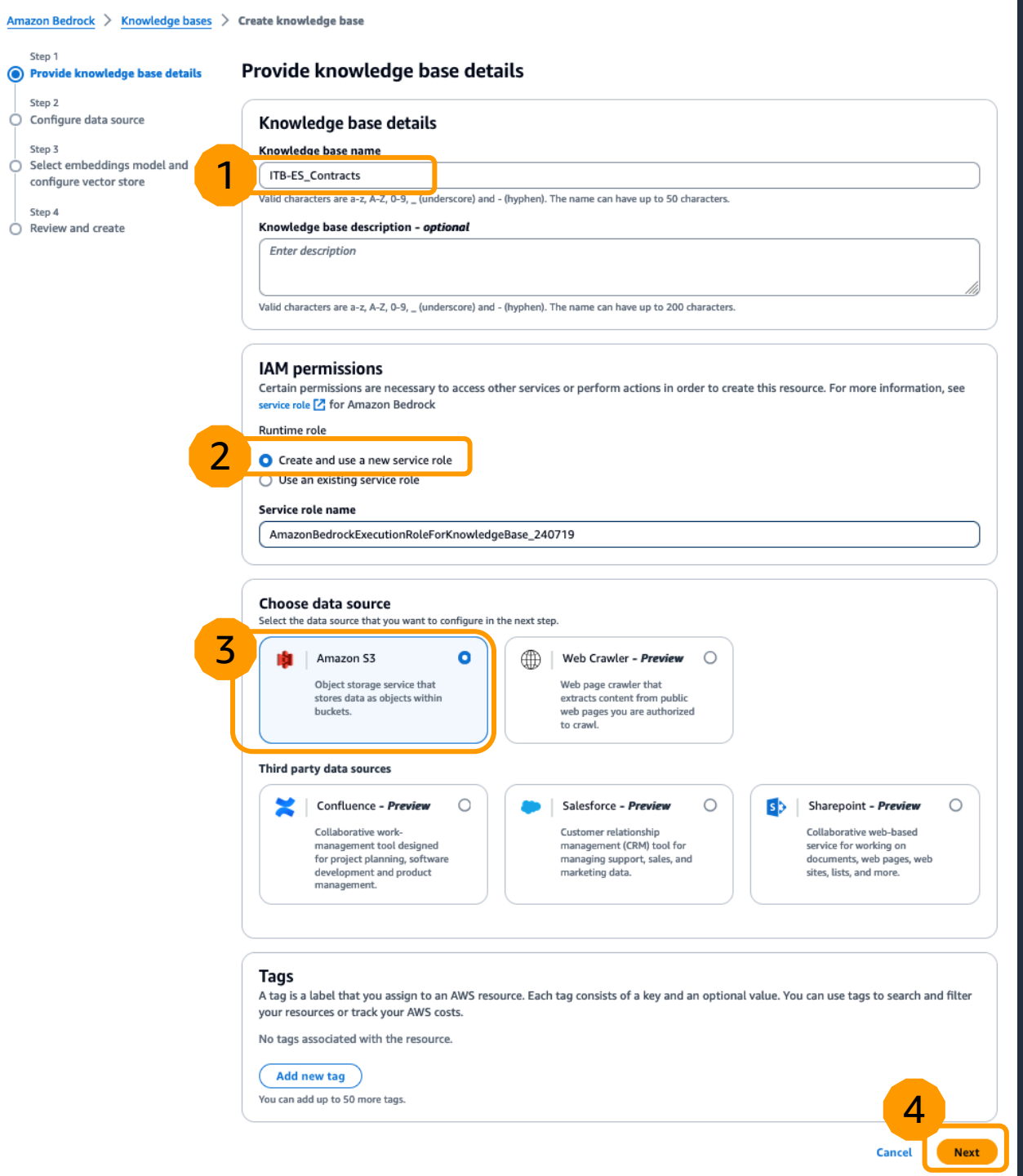
(1) Knowledge bases name 지정

(2) IAM 롤 생성

(이미 만든 롤이 있다면 기존 롤 선택 가능)

(3) Data source 는 S3 선택

(4) Next 클릭



3. Data source 구성 (1/2)

(1) Data source name 지정

(2) 문서가 업로드 된 S3 URI 선택

(Parsing Strategy를 사용하는 경우,

S3 URI 당 처리 가능 문서 **최대 100개**)

(3) 청킹 부분에서 Custom 선택

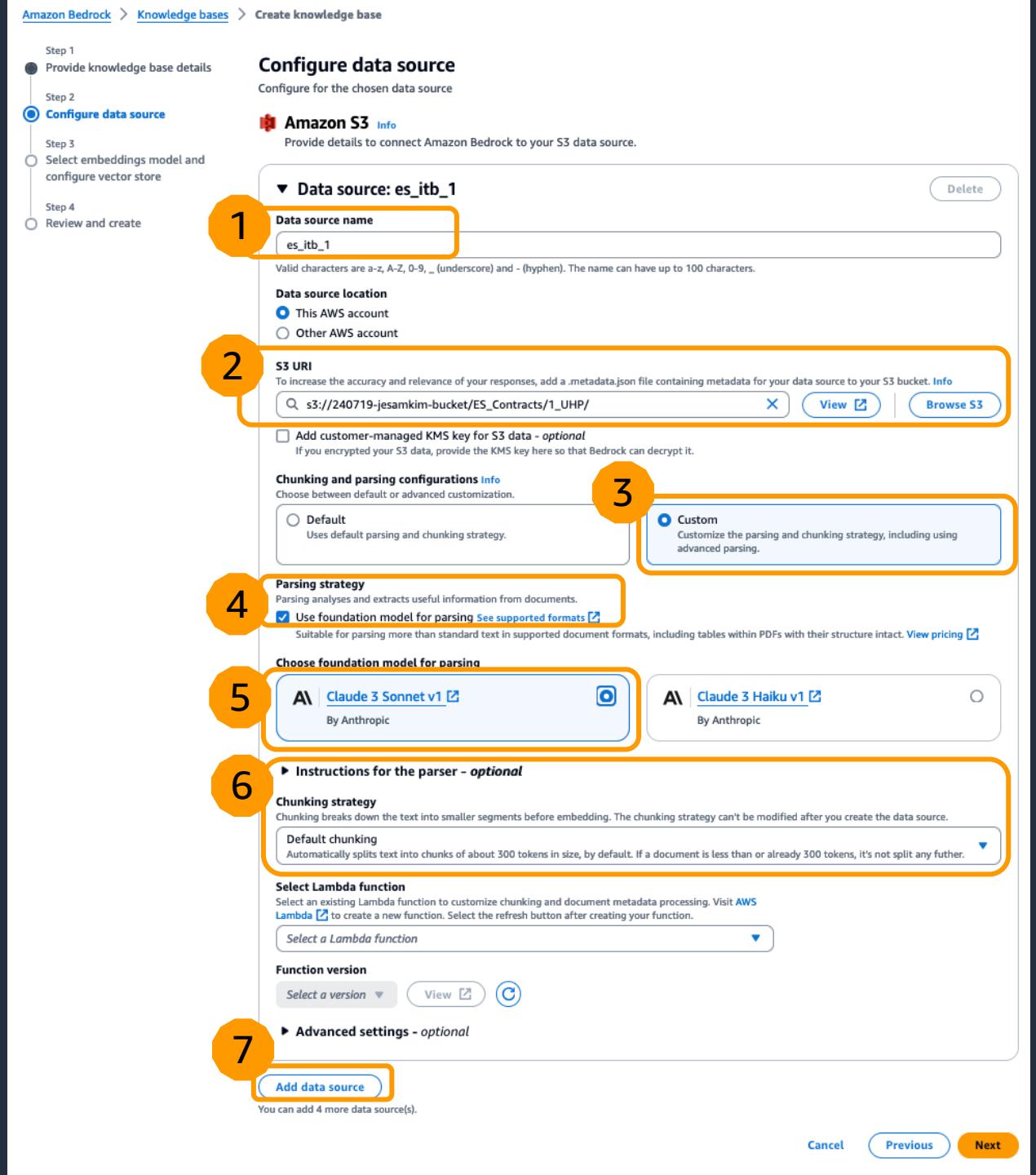
(4) Parsing strategy 사용으로 체크

(5) 문서 Parsing 부분에 사용할 모델 선택

→ Claude 3 Sonnet

(6) 선택사항 : Parsing strategy에 대한 프롬프트 수정 및 Chunking strategy는 바꿔볼 수 있음 (*문서 마지막 Appendix 참조*)

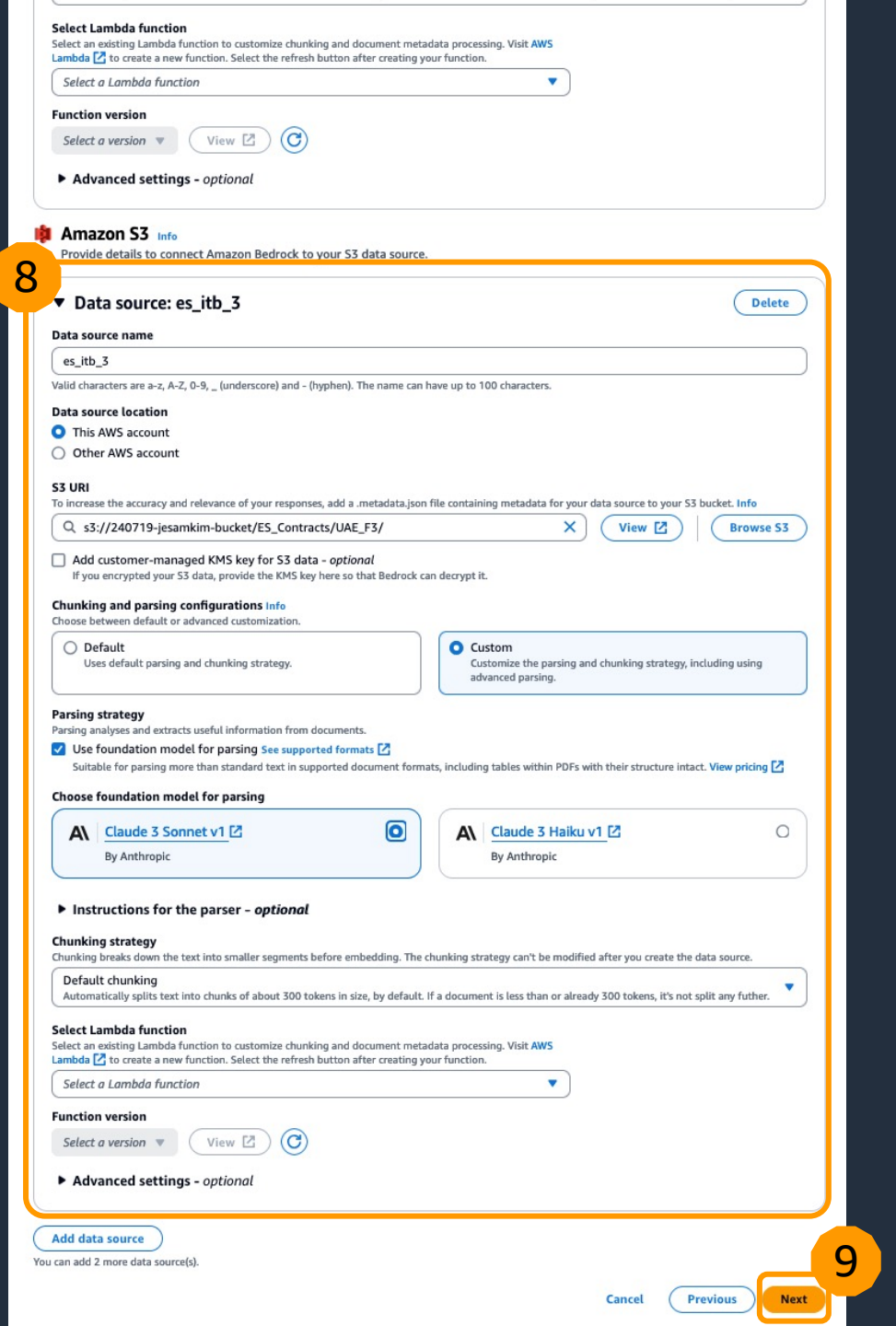
(7) 선택사항 : 추가 데이터 소스가 있다면 Add data source 선택



3. Data source 구성 (2/2)

(8) **선택사항 :** 추가 Data source 가 있다면 앞의 과정처럼 S3 URI를 지정하고 Chunking 구성 등은 동일하게 설정 합니다.

(9) Next 버튼 클릭



4. 임베딩 모델 선택

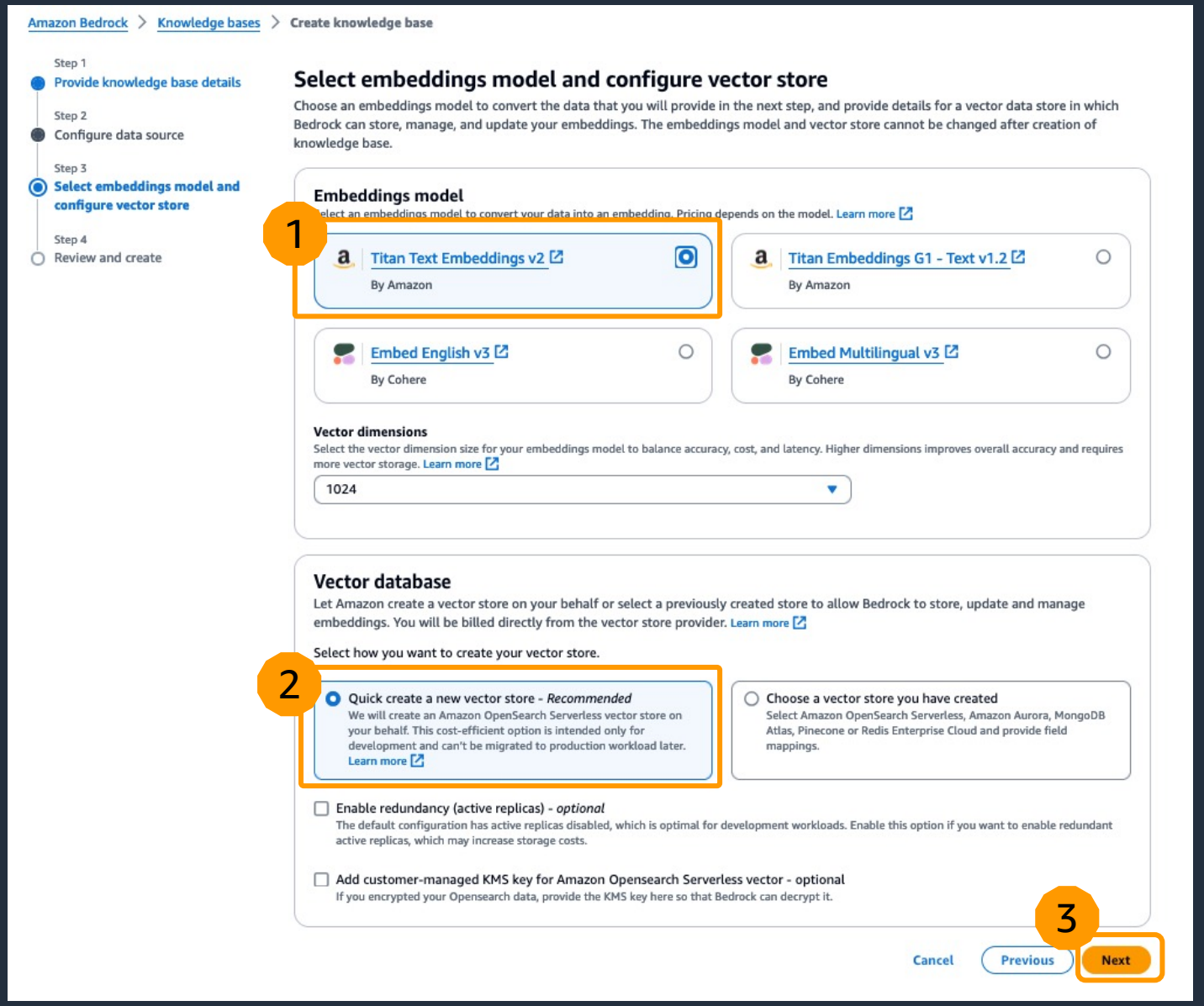
(1) 임베딩 모델 선택

→ 여기서는 Titan Emb v2 선택

(2) Vector database 부분에서는

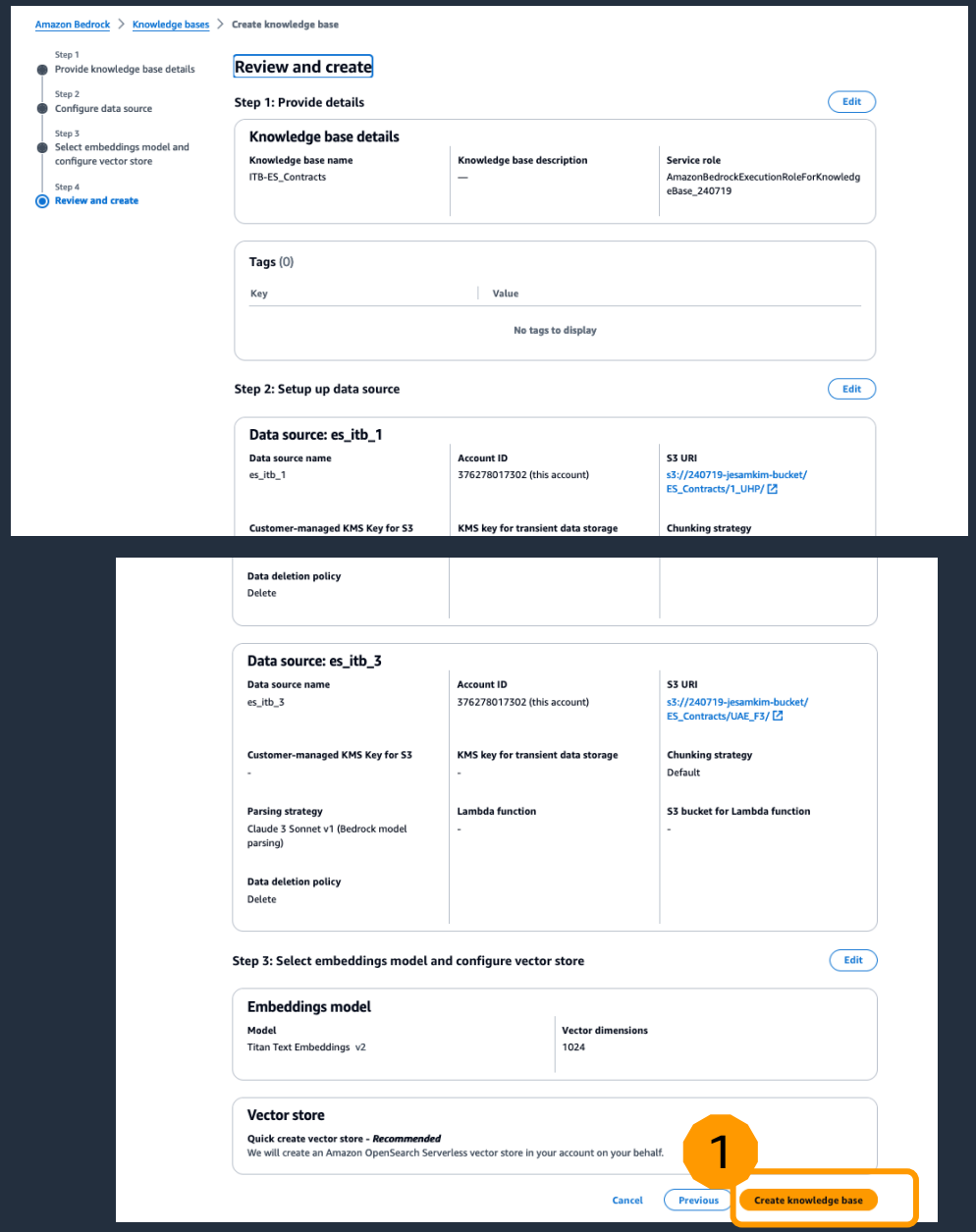
Quick create ~~ 를 선택 합니다.

(3) Next 클릭



5. 리뷰 & 생성

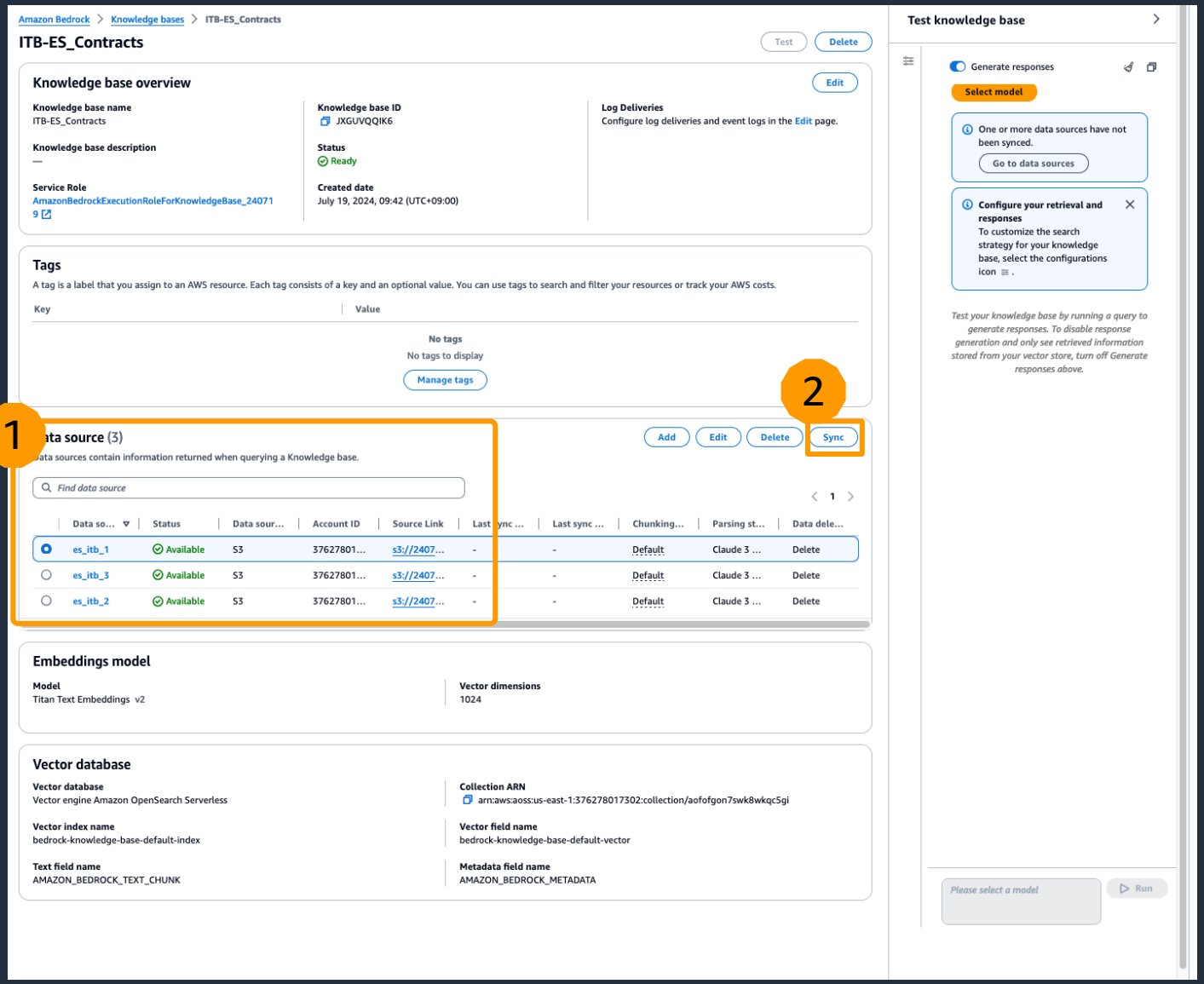
(1) 앞서 선택한 구성이 제대로 되었는지 확인하고 문제가 없으면 Create knowledge base 버튼을 클릭 합니다.



6. Data Source Sync (인덱싱) (1/2)

(1) 앞서 추가한 data source 항목이 보입니다.
(문서 갯수가 많아서 3개로 나누어서 등록했습니다;
Data source 1개 당 문서 최대 100개)

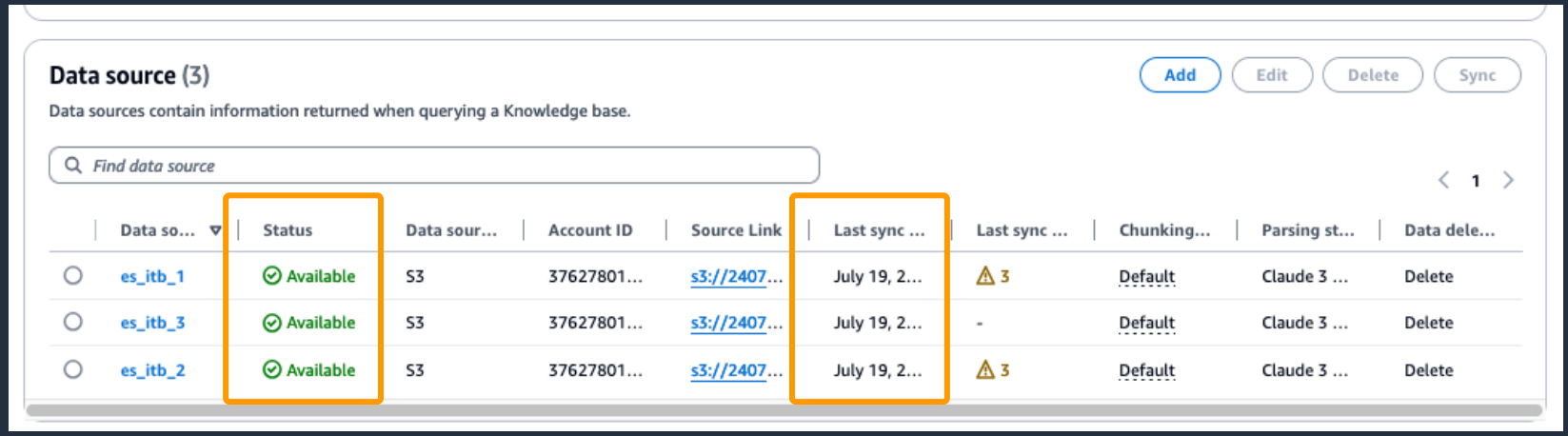
(2) Data source를 선택하고 Sync 버튼을 누릅니다. (인덱싱 시작)
나머지 문서도 순차적으로 Sync 합니다.



6. Data Source Sync (인덱싱) (2/2)

Sync 가 완료되면 Last sync에 시간이 표시 됩니다.

Status도 Available 인지 확인 입니다.



7. RAG 테스트 (1/2)

KB는 챗봇 UI가 제공되어, RAG 테스트를 콘솔에서 바로 해볼 수 있습니다.

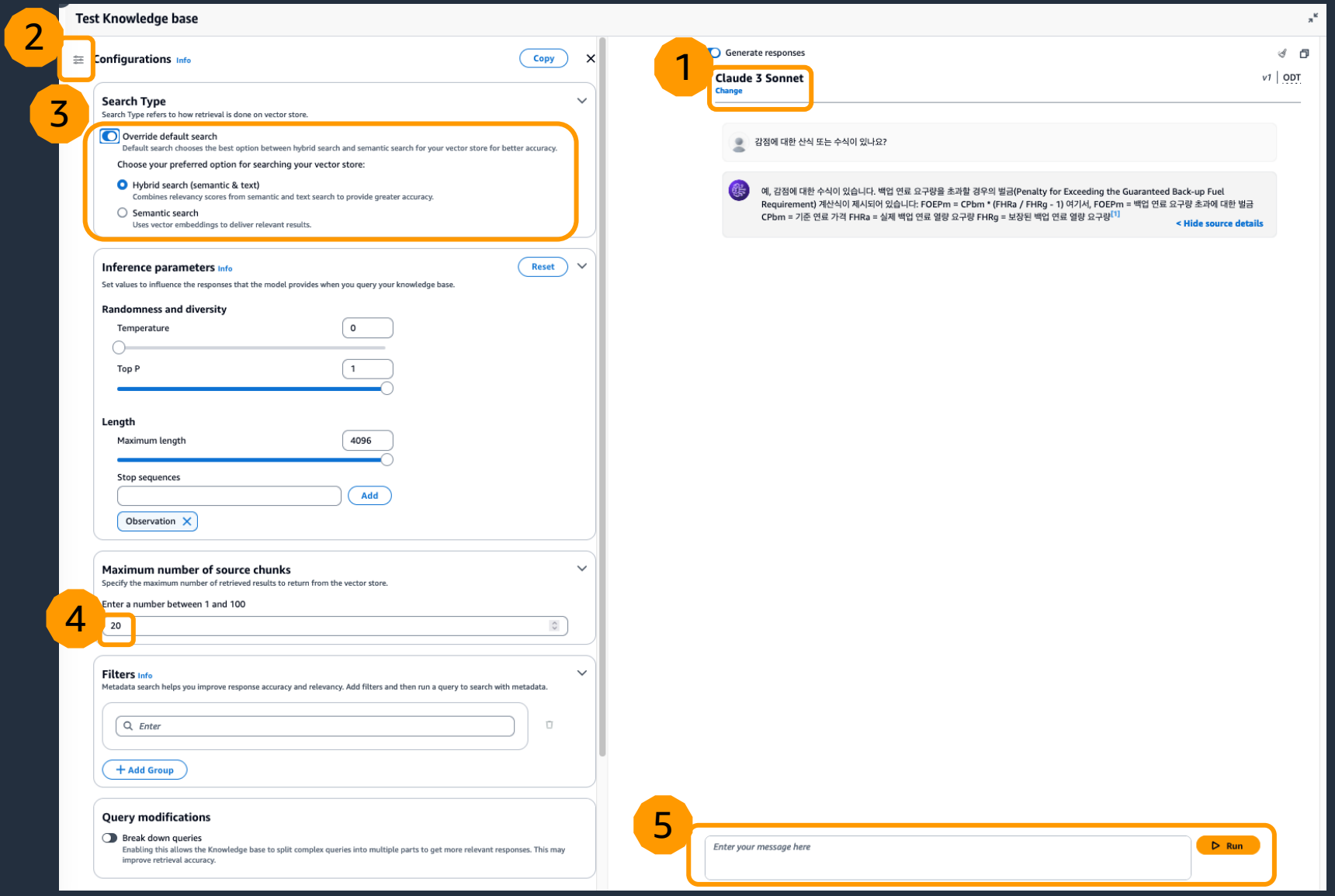
(1) 챗봇에 사용할 모델 선택

(2) Configurations 창 오픈

(3) Search Type 은 하이브리드 방식으로 변경

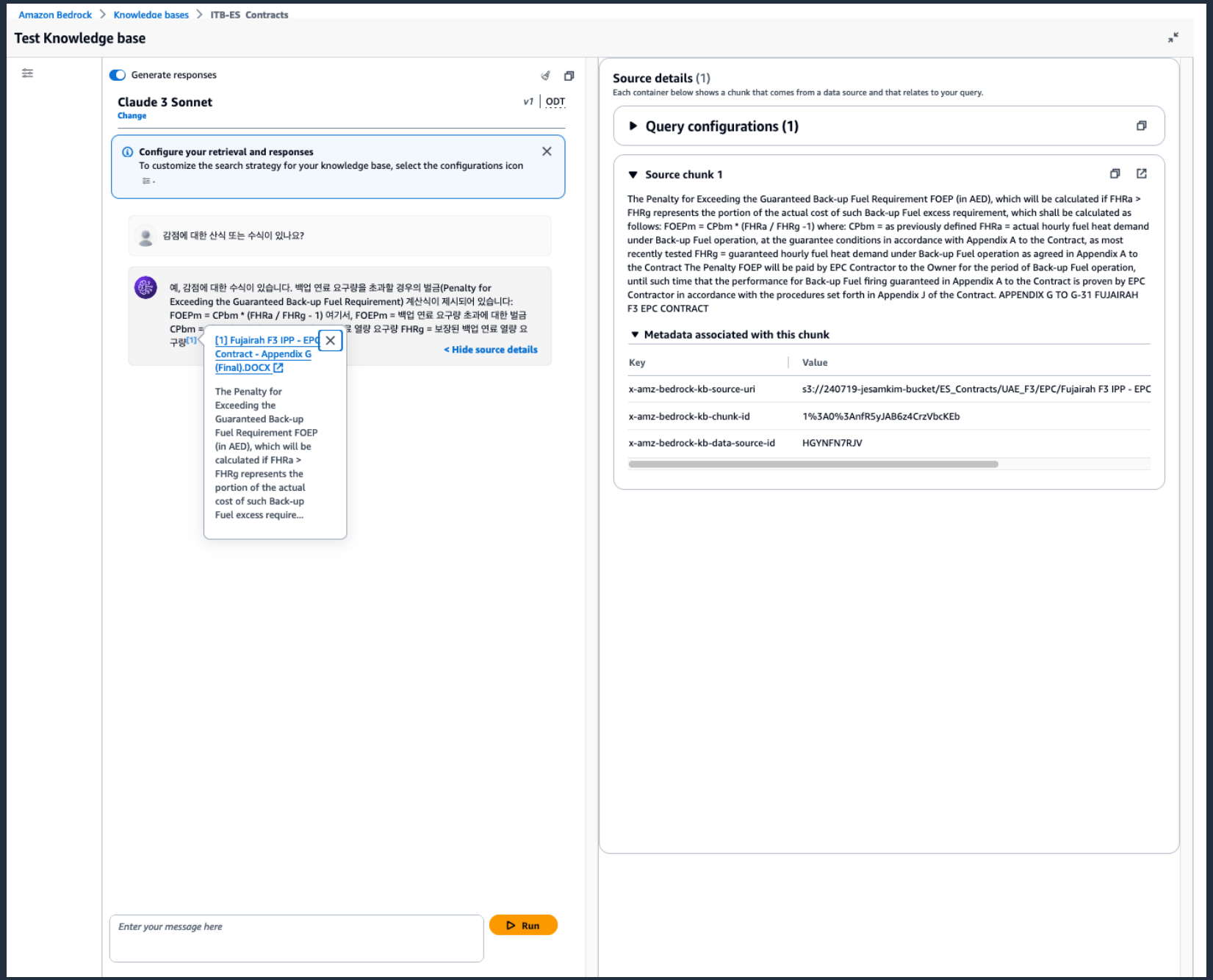
(4) 가져올 청크 갯수 조정

(5) 사용자 쿼리 입력



7. RAG 테스트 (2/2)

KB는 챗봇 UI에서는 사용자 쿼리에 대한 응답을 생성할때 사용한 Context를 함께 확인할 수 있습니다. (Ciatations)



Thank you!

Appendix

Instruction for the parser 프롬프트 (default)

```
Transcribe the text content from an image page and output in Markdown syntax (not code blocks). Follow these steps:

1. Examine the provided page carefully.

2. Identify all elements present in the page, including headers, body text, footnotes, tables, visulizations, captions, and page numbers, etc.

3. Use markdown syntax to format your output:
   - Headings: # for main, ## for sections, ### for subsections, etc.
   - Lists: * or - for bulleted, 1. 2. 3. for numbered
   - Do not repeat yourself

4. If the element is a visulization
   - Provide a detailed description in natural language
   - Do not transcribe text in the visualization after providing the description

5. If the element is a table
   - Create a markdown table, ensuring every row has the same number of columns
   - Maintain cell alignment as closely as possible
   - Do not split a table into multiple tables
   - If a merged cell spans multiple rows or columns, place the text in the top-left cell and output ' ' for other
   - Use | for column separators, |-|-| for header row separators
   - If a cell has multiple items, list them in separate rows
   - If the table contains sub-headers, separate the sub-headers from the headers in another row

6. If the element is a paragraph
   - Transcribe each text element precisely as it appears

7. If the element is a header, footer, footnote, page number
   - Transcribe each text element precisely as it appears

Output Example:

A bar chart showing annual sales figures, with the y-axis labeled "Sales ($Million)" and the x-axis labeled "Year". The chart has bars for 2018 ($12M), 2019 ($18M), 2020 ($8M), and 2021 ($22M).
Figure 3: This chart shows annual sales in millions. The year 2020 was significantly down due to the COVID-19 pandemic.

# Annual Report

## Financial Highlights

* Revenue: $40M
* Profit: $12M
* EPS: $1.25


| | Year Ended December 31, | |
| | 2021 | 2022 |
|-|-|-|
| Cash provided by (used in): | | |
| Operating activities | $ 46,327 | $ 46,752 |
| Investing activities | (58,154) | (37,601) |
| Financing activities | 6,291 | 9,718 |

Here is the image.
```

선택 가능한 Chunking strategy

Chunking strategy

Chunking breaks down the text into smaller segments before embedding. The chunking strategy can't be modified after you create the data source.

Default chunking
Automatically splits text into chunks of about 300 tokens in size, by default. If a document is less than or already 300 tokens, it's not split any futher.

Default chunking ✓
Automatically splits text into chunks of about 300 tokens in size, by default. If a document is less than or already 300 tokens, it's not split any futher.

Fixed-size chunking
Splits text into your set approximate token size.

Hierarchical chunking
Organizes text chunks (nodes) into hierarchical structures of parent-child relationships. Each child node includes a reference to its parent node.

Semantic chunking
Organizes text chunks or groups of sentences by how semantically similar they are to each other.

No chunking
Suitable for documents that are already pre-processed or text split into separate files without any further chunking necessary.