

## What is Document AI?

Think of Document AI as a super-smart assistant that can read and understand your documents quickly and accurately. This amazing feature from Snowflake uses a special technology called Arctic-TILT to pick out important information from all sorts of documents. Whether it's a long paragraph, a logo, or even a handwritten signature, Document AI can handle it all. It's perfect for organizing documents like invoices or financial statements into clear, easy-to-read tables.

## Why Should You Care About Document AI?What is Document AI?

**Smart Extraction:** Document AI can find and extract the right information even from documents it has never seen before. This is called **zero-shot** extraction, and it works like magic for your data.

**Customizable:** You can train the model with your own documents to make it work even better for your specific needs. Plus, this custom model is private and won't be shared with anyone else.

## When to Use Document AI

**Organize Your Data:** Turn messy, unstructured data from documents into neat tables.

**Automate Your Work:** Set up continuous processing for new documents of the same type.

**Team Collaboration:** Business experts can set up the model, while data engineers create pipelines using SQL for ongoing document processing.

## How Does Document AI Work?

**Easy-to-Use Interface:** Create, test, and improve your Document AI model with ease.

**Model Build:** This represents a specific type of document or use case, like extracting details from invoices. It includes the model, the data to be pulled out, and documents to train and test the model.

**Extracting Query:** Use a simple query to pull information from documents and set up continuous processing with streams and tasks.

Want to know more? Dive into how **Document AI** can transform the way you handle documents and make your work life much easier!

## Create a document processing pipeline with Document AI

### *- Setup the required objects and privileges*

```
--Create a database and schema in which to create a Document AI model build:
CREATE DATABASE doc_ai_db;
CREATE SCHEMA doc_ai_db.doc_ai_schema;

--Create custom role doc_ai_role

USE ROLE ACCOUNTADMIN;
```

```

CREATE ROLE doc_ai_role;
--Grant the SNOWFLAKE.DOCUMENT_INTELLIGENCE_CREATOR database role to the
doc_ai_role role:

    GRANT DATABASE ROLE SNOWFLAKE.DOCUMENT_INTELLIGENCE_CREATOR TO ROLE
doc_ai_role;

--Grant warehouse usage and operating privileges to the doc_ai_role role:

    GRANT USAGE, OPERATE ON WAREHOUSE <your_warehouse> TO ROLE doc_ai_role;

--Grant the privileges to use the database and schema you created to the
doc_ai_role:

    GRANT USAGE ON DATABASE doc_ai_db TO ROLE doc_ai_role;
    GRANT USAGE ON SCHEMA doc_ai_db.doc_ai_schema TO ROLE doc_ai_role;

--Grant the create stage privilege on the schema to the doc_ai_role role to
store the documents for extraction:

    GRANT CREATE STAGE ON SCHEMA doc_ai_db.doc_ai_schema TO ROLE
doc_ai_role;

--Grant the privilege to create model builds (instances of the
DOCUMENT_INTELLIGENCE class) to the doc_ai_role role:

    GRANT CREATE SNOWFLAKE.ML.DOCUMENT_INTELLIGENCE ON SCHEMA
doc_ai_db.doc_ai_schema TO ROLE doc_ai_role;

--Grant the privileges required to create a processing pipeline using streams
and tasks to the doc_ai_role role:

    GRANT CREATE STREAM, CREATE TABLE, CREATE TASK, CREATE VIEW ON SCHEMA
doc_ai_db.doc_ai_schema TO ROLE doc_ai_role;
    GRANT EXECUTE TASK ON ACCOUNT TO ROLE doc_ai_role;

--Grant the doc_ai_role to tutorial user for use in the next steps of the
tutorial:

    GRANT ROLE doc_ai_role TO USER <your_user_name>;

```

## - Prepare a Document AI model build

- Create a Document AI model build.
- Upload documents to test the Document AI model build.

### New Build

Document AI build is a collection of (a) document samples for a specific type of document (e.g. a set of 20 invoices) (b) natural language questions to extract specific values (e.g. invoice amount, date) from these documents, and (c) an AI model that gets automatically built to extract these values across many documents (e.g. 1000s of invoices a day). [Learn more](#)

**Build name**

**Location ⓘ**

DOC\_AI\_DB

▼

DOC\_AI\_SCHEMA


▼

**Description (optional)**

Provide a description that would reflect the main use case of preparing the build

Cancel

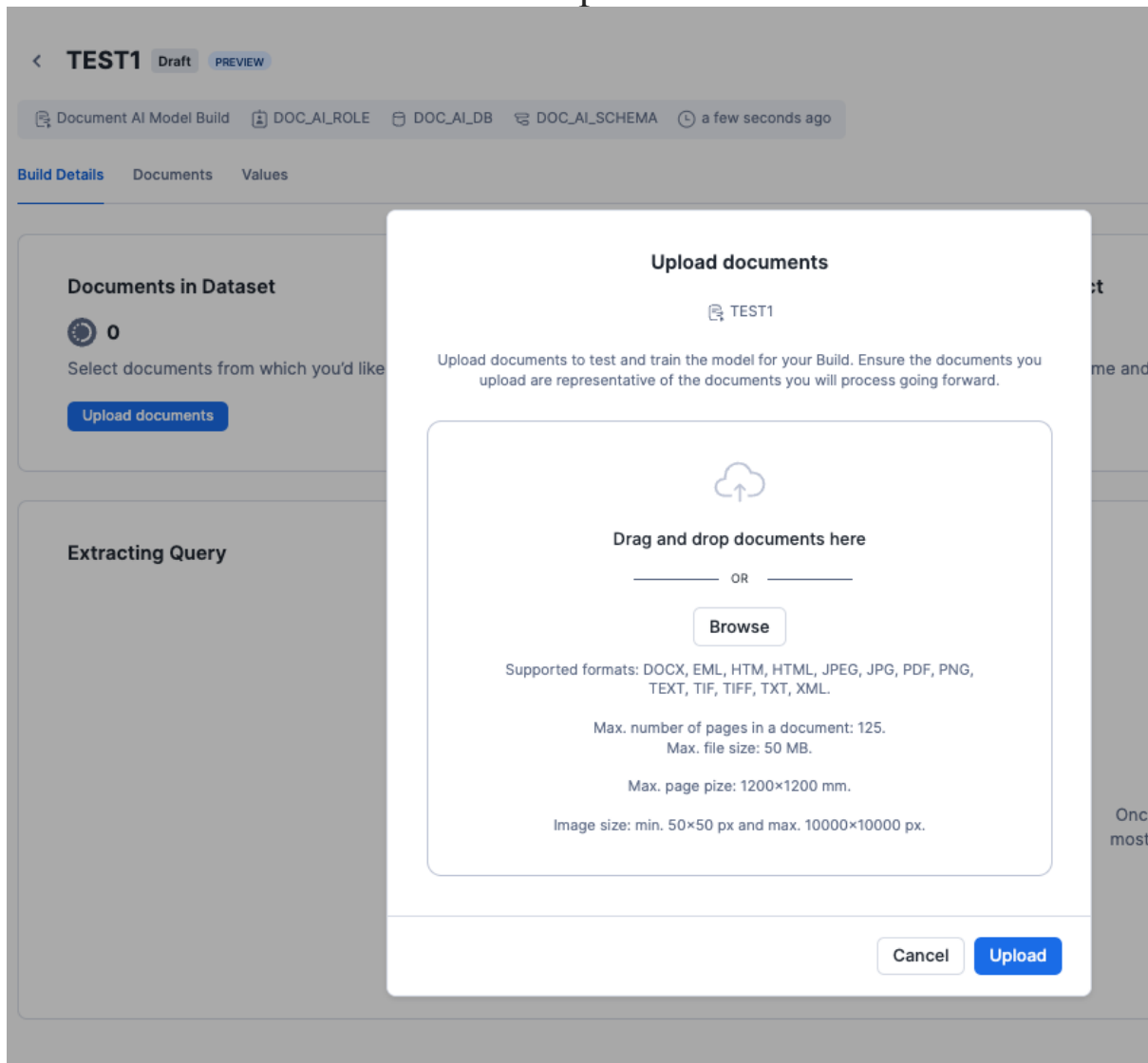
Create



### Welcome to Document AI Model Builds

Document AI is a Snowflake feature that uses a large language model (LLM) to extract data from documents. [Learn more](#)

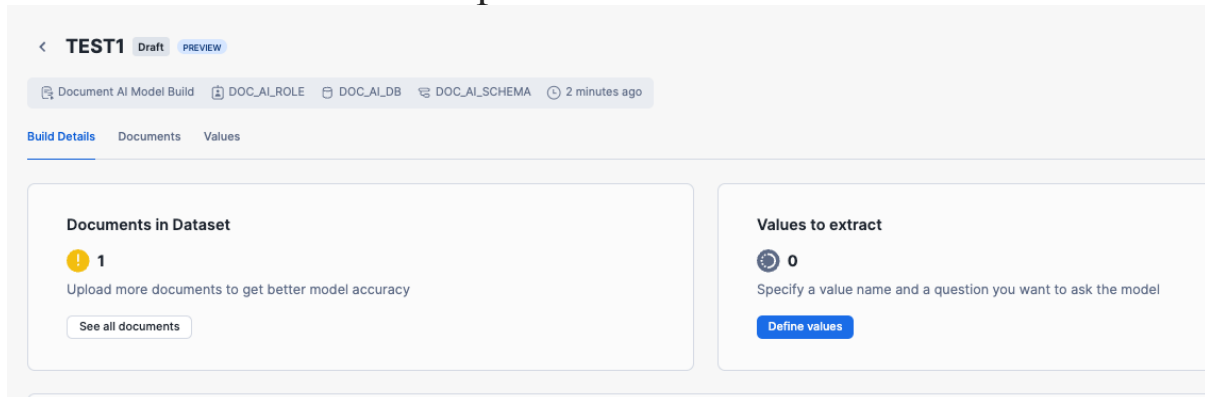
To create a Document AI model build, sign in to Snowsight, go to AI & ML » Document AI, choose a warehouse, and click + Build. Name your model build and select the location, then hit Create. To upload documents to this model, download the necessary [files](#), unzip them, and in the model build's Build Details tab, select Upload documents. Choose the documents and click Upload. That's it!



## - Define data values and review the results

- Define data values to extract by asking the model questions in natural language.

- Review results by confirming or correcting the answers that the model provided



To define values for the Document AI model build:

1. In the TEST1 model build, select the **Build Details** tab.
2. Select **Define values**.
3. In the **Documents review** view, select + **Value**.
4. For each document, enter the following pairs of value names and questions:
  - `inspection_date`: What is the inspection date?
  - `inspection_grade`: What is the grade?
  - `inspector`: Who performed the inspection?
  - `list_of_units`: What are all the units?
5. For each document and data value, review the answers that the model provides:

- If the answer is correct, select the checkmark.
- If the answer is incorrect, enter the correct value manually.

uments review

SkiGear Co.  
1661 Mesa Drive  
Las Vegas, Nevada  
123-555-5555

**EQUIPMENT INSPECTION**

MACHINE	SERIAL NUMBER	INSPECTION GRADE
Injection Molder	SGMM-12345	PASS

**INSPECTION SUMMARY**

During the inspection of the injection molder, everything appeared to be in good working order. The technician thoroughly examined the machine, including its components and operating systems, and found no issues. All safety measures were also confirmed to be functional and up to standard. The technician recommended regular maintenance and upkeep to ensure the machine's continued optimal performance. Overall, the inspection yielded no concerns or defects, and the machine was deemed fit for production.

Injection Unit	Good	✓
Mold Clamping Unit	Good	✓
Hydraulic System	Good	✓
Temperature Control System	Good	✓
Ejector System	Good	✓
Lubrication System	Good	✓
Safety Devices	Good	✓
Control Software	Good	✓

Emily Johnson  
Inspected by

2023-01-01  
Date

Manual\_2023-01-01.pdf

Values to extract

inspection\_date What is the inspection date?

0.98 2023-01-01 ✓

inspection\_grade What is the grade?

0.53 PASS ✓

inspector Who performed the inspection?

0.94 Emily Johnson ✓

list\_of\_units What are all the units?

0.92 Injection Unit ✓

0.93 Mold Clamping Unit ✓

0.93 Hydraulic System ✓

0.93 Temperature Control System ✓

0.93 Ejector System ✓

0.93 Lubrication System ✓

0.93 Safety Devices ✓

0.93 Control Software ✓

Add value

## - Publish a Document AI model build

To publish the model build, do the following:

1. In the TEST1 model build, select the **Build Details** tab.
2. Under **Model accuracy**, select **Publish version**.
3. In the dialog that appears, select **Publish** to confirm.

TEST1 Draft PREVIEW

Document AI Model Build DOC\_AI\_ROLE DOC\_AI\_DB DOC\_AI\_SCHEMA 6 minutes ago

Build Details Documents Values

**Documents in Dataset**

1

Upload more documents to get better model accuracy

See all documents

**Values to extract**

4

Number of values defined to be extracted from the Dataset

See all values

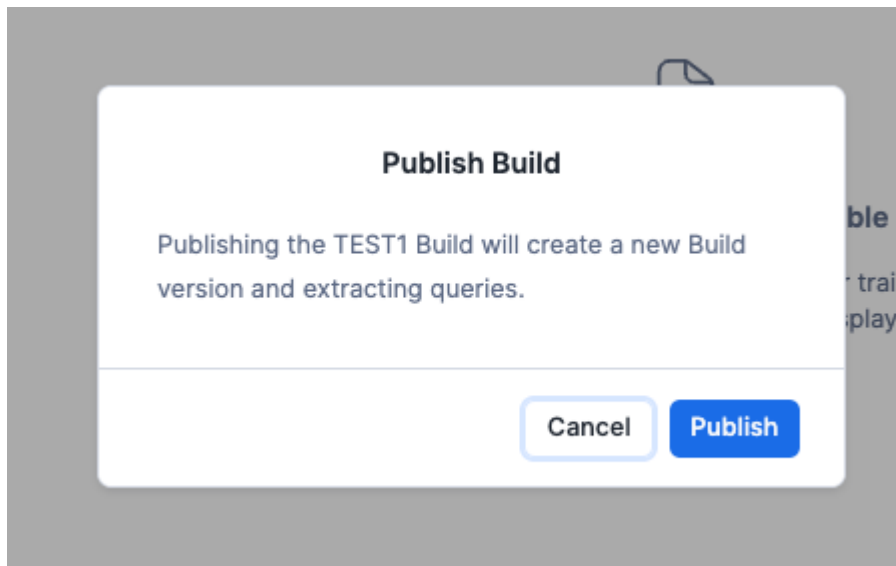
**Model accuracy**

1.00 build version: not available

Accuracy describes how often the model provides a correct answer (up to 1.00)

Publish version Train model

Extracting Query



## - Create a document processing pipeline

- Set up the pipeline using streams and tasks.
- Upload new documents to an internal stage.
- View the extracted information.

```
--Create an internal my_pdf_stage stage to store the documents:
CREATE OR REPLACE STAGE my_pdf_stage DIRECTORY = (ENABLE = TRUE)

ENCRYPTION = (TYPE = 'SNOWFLAKE_SSE');

--Create a my_pdf_stream stream on a my_pdf_stage stage:
CREATE STREAM my_pdf_stream ON STAGE my_pdf_stage;

--Refresh
ALTER STAGE my_pdf_stage REFRESH;

--Specify the database and schema:

USE DATABASE doc_ai_db;

USE SCHEMA doc_ai_schema;

--Create a pdf_reviews table to store the information
CREATE OR REPLACE TABLE pdf_reviews (
  file_name VARCHAR,
  file_size VARIANT,
```



```

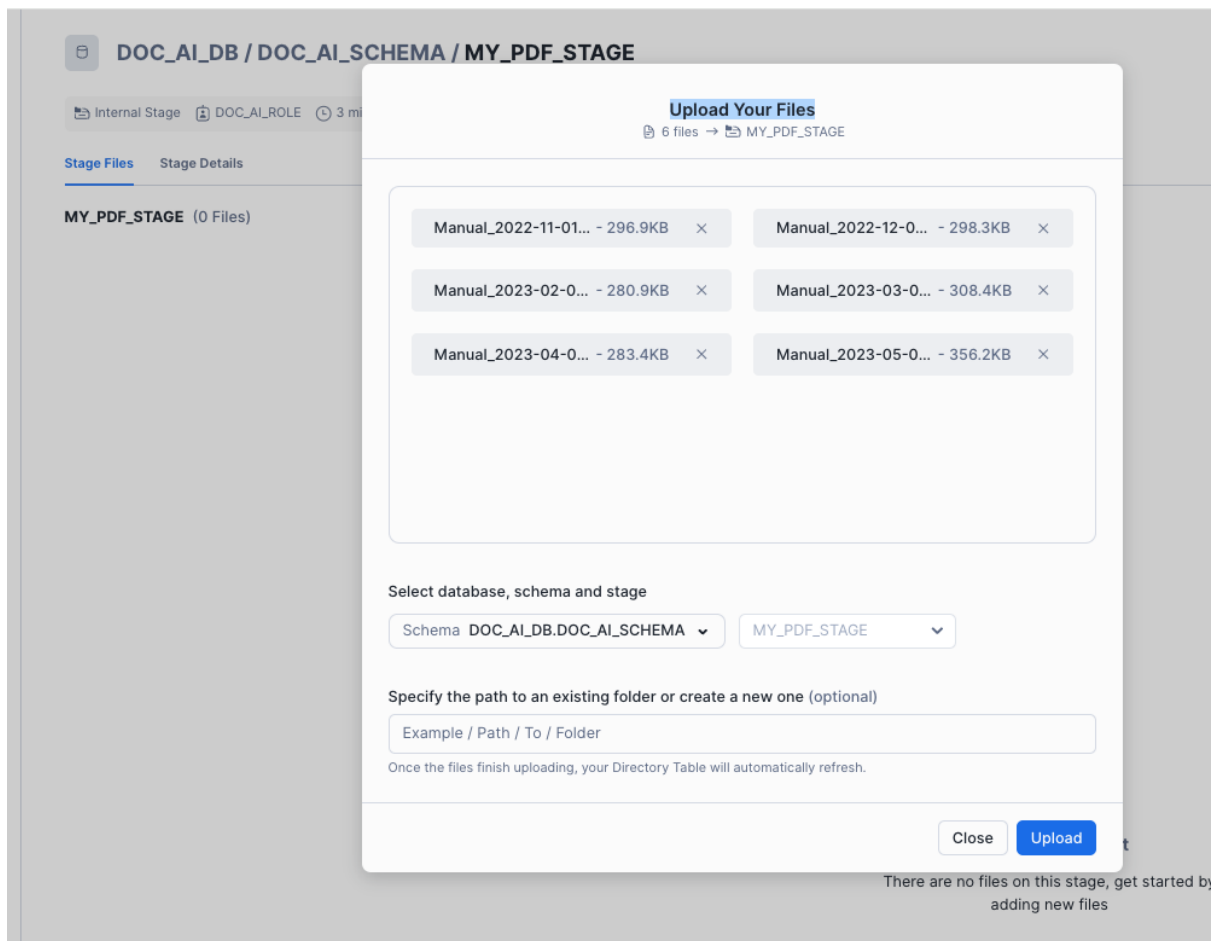
        last_modified VARCHAR,
        snowflake_file_url VARCHAR,
        json_content VARCHAR
    );

--Create a load_new_file_data task to process new documents in the stage
CREATE OR REPLACE TASK load_new_file_data
    WAREHOUSE = doc_ai_wh
    SCHEDULE = '1 minute'
    COMMENT = 'Process new files in the stage and insert data into the
pdf_reviews table.'
    WHEN SYSTEM$STREAM_HAS_DATA('my_pdf_stream')
    AS
    INSERT INTO pdf_reviews (
        SELECT
            RELATIVE_PATH AS file_name,
            size AS file_size,
            last_modified,
            file_url AS snowflake_file_url,
            TEST1!PREDICT(GET_PREIGNED_URL('@my_pdf_stage', RELATIVE_PATH), 1)
AS json_content
        FROM my_pdf_stream
        WHERE METADATA$ACTION = 'INSERT'
    );

--Start the newly created task:
ALTER TASK load_new_file_data RESUME;

```

To upload new documents, first, download the [zip](#) files with the needed documents to your computer. Then, unzip the files, which are in PDF format. In Snowsight, go to Data » Databases. Choose the doc\_ai\_db database, the doc\_ai\_schema, and the my\_pdf\_stage stage. Click on + Files, then select the files you downloaded. Finally, click Upload. That's it! Your documents are now uploaded and ready to use.



```
--After uploading the documents to the stage, view the information
extracted
SELECT * FROM pdf_reviews;

--Create a pdf_reviews_2 table to analyze the extracted information in
separate columns:

CREATE OR REPLACE TABLE doc_ai_db.doc_ai_schema.pdf_reviews_2 AS (
  WITH temp AS (
    SELECT
      RELATIVE_PATH AS file_name,
      size AS file_size,
      last_modified,
      file_url AS snowflake_file_url,
      TEST1!PREDICT(get_presigned_url('@my_pdf_stage', RELATIVE_PATH), 1)
    AS json_content
    FROM directory(@my_pdf_stage)
  )

  SELECT
    file_name,
    file_size,
    last_modified,
    snowflake_file_url,
    json_content:documentMetadata.ocrScore::FLOAT AS ocrScore,
    f.value:score::FLOAT AS inspection_date_score,
```

```

        f.value:value::STRING AS inspection_date_value,
        g.value:score::FLOAT AS inspection_grade_score,
        g.value:value::STRING AS inspection_grade_value,
        i.value:score::FLOAT AS inspector_score,
        i.value:value::STRING AS inspector_value,
        ARRAY_TO_STRING(ARRAY_AGG(j.value:value::STRING), ', ') AS
list_of_units
FROM temp,
    LATERAL FLATTEN(INPUT => json_content:inspection_date) f,
    LATERAL FLATTEN(INPUT => json_content:inspection_grade) g,
    LATERAL FLATTEN(INPUT => json_content:inspector) i,
    LATERAL FLATTEN(INPUT => json_content:list_of_units) j
GROUP BY ALL
);

--View the output:
SELECT * FROM pdf_reviews_2;

```

```

41 SELECT * FROM pdf_reviews;
42
43
44 CREATE OR REPLACE TABLE doc_ai_db.doc_ai_schema.pdf_reviews_2 AS (
45 WITH temp AS (
46 SELECT
47     RELATIVE_PATH AS file_name,
48     size AS file_size,
49     last_modified,
50     file_url AS snowflake_file_url,
51     TEST1(PREDICT(get_presigned_url('@my_pdf_stage', RELATIVE_PATH), 1) AS json_content
52 FROM directory('@my_pdf_stage')
53 )
54
55 SELECT
56     file_name,
57     file_size,
58     last_modified,
59     snowflake_file_url,
60     json_content::documentMetadata.ocrScore::FLOAT AS ocrScore,
61     f.value:score::FLOAT AS inspection_date_score,
62     f.value:value::STRING AS inspection_date_value,
63     g.value:score::FLOAT AS inspection_grade_score,
64     g.value:value::STRING AS inspection_grade_value,
65     i.value:score::FLOAT AS inspector_score,
66     i.value:value::STRING AS inspector_value,
67     ARRAY_TO_STRING(ARRAY_AGG(j.value:value::STRING), ', ') AS list_of_units
68 FROM temp,
69     LATERAL FLATTEN(INPUT => json_content:inspection_date) f,
70     LATERAL FLATTEN(INPUT => json_content:inspection_grade) g,
71     LATERAL FLATTEN(INPUT => json_content:inspector) i,
72     LATERAL FLATTEN(INPUT => json_content:list_of_units) j
73 GROUP BY ALL
74 );
75
76 SELECT * FROM pdf_reviews_2;
77

```

FILE_NAME	FILE_SIZE	LAST_MODIFIED	SNOWFLAKE_FILE_URL	OCRSORE	INSPECTION_DATE_SCORE	INSPECTION_DATE_VALUE	INSPECTION_GRADE_SCORE	INSPECTION_GRADE_VALUE	INSPECTOR_SCORE	IN
1 Manual_2023-03-01.pdf	315754	2024-06-07 15:46:42.000 -0500	https://unigrou.us-east-1-	0.978	0.894	2023-03-01	0.363	FAIL	0.945	En
2 Manual_2023-04-01.pdf	290188	2024-06-07 15:46:42.000 -0500	https://unigrou.us-east-1-	0.976	0.919	2023-04-01	0.532	PASS	0.945	En
3 Manual_2023-05-01.pdf	364722	2024-06-07 15:46:43.000 -0500	https://unigrou.us-east-1-	0.977	0.883	2023-05-01	0.581	PASS	0.944	En
4 Manual_2022-12-01.pdf	305429	2024-06-07 15:46:41.000 -0500	https://unigrou.us-east-1-	0.974	0.814	2022-12-01	0.418	FAIL	0.947	En
5 Manual_2022-11-01.pdf	304014	2024-06-07 15:46:40.000 -0500	https://unigrou.us-east-1-	0.972	0.876	2022-11-01	0.58	PASS	0.945	En
6 Manual_2023-02-01.pdf	287596	2024-06-07 15:46:42.000 -0500	https://unigrou.us-east-1-	0.977	0.894	2023-02-01	0.545	PASS	0.934	En

You've learned to set up a place to keep your documents safe, created steps to organize them effectively, and easily uploaded them for processing. Plus, you could see all the important details neatly arranged in a table. Now, you're all set to manage your documents smoothly, making your work life a whole lot easier!

<https://docs.snowflake.com/en/user-guide/snowflake-cortex/document-ai/tutorials/create-processing-pipelines#introduction>