

Cloudera AI

Using AI Studios

Date published: 2020-07-16

Date modified: 2025-10-31

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

AI Studios Overview (Technical Preview).....	4
Using RAG Studio.....	4
RAG Studio Overview.....	4
Key features for RAG Studio.....	5
Using RAG Studio.....	6
Use cases for RAG Studio.....	8
Using Fine Tuning Studio.....	9
Fine Tuning Studio Overview.....	10
Key Features for Fine Tuning Studio.....	10
Using Fine Tuning Studio.....	11
Use case for Fine Tuning Studio - Event ticketing support.....	12
Using Synthetic Data Studio.....	16
Synthetic Data Studio Overview.....	16
Synthetic Data Studio use cases.....	17
Key Features of Synthetic Data Studio.....	17
Launching Synthetic Data Studio within a project.....	18
Generating synthetic data for fine-tuning models.....	19
Evaluating generated datasets for fine-tuning LLMs.....	21
Managing generated datasets.....	22
Use case: Data generation for ticketing system using Synthetic Data Studio.....	22
Generating synthetic data for a ticketing use case using the Supervised Fine-Tuning workflow.....	23
Evaluating the generated dataset.....	25
Using Agent Studio.....	27
Agent Studio Overview.....	28
Key Features of Agent Studio.....	28
Use Cases of Agent Studio.....	29
Launching Agent Studio within a Project.....	29
Use case: Sequential Idea Generation Workflow.....	33

AI Studios Overview (Technical Preview)

Cloudera AI Studios is a comprehensive suite of low-code tools designed to simplify the development, customization, and deployment of generative AI solutions within enterprises. This suite empowers organizations to operationalize AI workflows quickly and efficiently by leveraging real-time enterprise data. It provides scalable, cost-effective, and trustworthy AI applications, fostering seamless collaboration between business and IT teams.



Note: This feature is in Technical Preview and not recommended for production deployments. Cloudera recommends that you try this feature in test or development environments.

Key Components of Cloudera AI Studios:

- **RAG Studio:**
 - Purpose: Accelerate the creation of secure, enterprise-grade Retrieval-Augmented Generation (RAG) applications and simple chatbots.
 - Capabilities: Enables organizations to build advanced, context-aware AI chatbots that utilize enterprise data while maintaining data security and compliance.
- **Fine-tuning Studio:**
 - Purpose: Simplify the customization and optimization of large language models (LLMs).
 - Capabilities: Allows users to fine-tune pre-trained models to suit specific business requirements, eliminating the need for extensive technical expertise or computational resources.
- **Synthetic Data Studio:**
 - Purpose: Generate scalable, privacy-compliant synthetic datasets for enterprise use cases.
 - Capabilities: Provides tools to create data that mimics real-world datasets while ensuring privacy and compliance, supporting AI model training and testing with minimal risk.
- **Agent Studio:**
 - Purpose: Design and deploy sophisticated multi-agent AI workflows tailored for enterprise automation.
 - Capabilities: Facilitates the orchestration of AI agents to perform complex tasks collaboratively, streamlining operations, and enhancing efficiency across the organization.

By leveraging Cloudera AI Studios, enterprises can rapidly build and deploy AI-powered applications, enabling innovation, improving decision-making, and driving business value, all while maintaining scalability, compliance, and trustworthiness in their AI initiatives.

Using RAG Studio

RAG Studio allows you to leverage the latest advancements in AI and machine learning while retaining full control over your data and workflows.

About this task

RAG Studio Overview

Retrieval-Augmented Generation (RAG) Studio is a no-code application for creating RAG chatbots that combine the power of retrieval and generation to provide accurate and efficient responses.

Using the Cloudera platform, you can build and deploy RAG chatbots in minutes, without requiring extensive technical expertise. Designed for accessibility, RAG Studio bridges the gap between business and IT teams, fostering collaboration on AI projects. The Retrieval-Augmented Generation studio features a secure, context-aware chatbot that leverages enterprise documents and real-time data ingestion to deliver accurate and efficient responses.

Built on open-source solutions, RAG Studio provides a flexible and customizable framework for developing and deploying chatbots. It allows you to leverage the latest advancements in AI and machine learning while retaining full control over your data and workflows.

RAG Studio is part of the Cloudera AI product portfolio and offers the added benefit of being deployable on premises, providing a secure and controlled environment for developing and deploying AI-powered chatbots. This allows you to integrate with existing data infrastructure and workflows, while also maintaining control over your data.

Key features for RAG Studio

RAG Studio allows you to create an application with a conversational interface (chatbot) that can be used to converse with a connected large language model, and comes with the functionality to enable retrieval-augmented generation to improve the performance and accuracy of the model's responses.

Core Features

- **No-Code Application:** RAG Studio is a no-code application, empowering users to create and deploy RAG chatbots without requiring extensive technical expertise.
- **Retrieval-Augmented Generation:** RAG Studio combines the power of retrieval and generation to deliver accurate and efficient responses, leveraging the strengths of both approaches.
- **Secure and Context-Aware:** Cloudera's chatbot is designed to be secure and context-aware, utilizing enterprise documents and real-time data ingestion to provide precise and relevant responses. Each answer synthesized from your enterprise documents will be automatically scored for faithfulness and relevance and provide reference documents, for easy and reliable human use.
- **Open-Source and Customizable:** RAG Studio is built on open-source solutions, offering a flexible and customizable framework for developing and deploying chatbots that meet your unique needs.

Deployment and Integration

- **On-Premises Deployment:** RAG Studio can be deployed on premises, providing a secure and controlled environment for developing and deploying AI-powered chatbots.
- **Integration with Existing Infrastructure:** RAG Studio seamlessly integrates with existing data infrastructure and workflows, allowing you to maintain control over your data while leveraging the power of AI.

Accessibility and Collaboration

- **Accessibility:** RAG Studio is designed for accessibility, bridging the gap between business and IT teams and fostering collaboration on AI projects.

Advanced Capabilities

- **Optimized with Vector Search:** Our platform is optimized with vector search, enabling fast and efficient retrieval of relevant information.
- **Automatic Evaluation:** Our platform includes automatic evaluation, allowing you to assess the quality and accuracy of your chatbot's responses.
- **Scalable NiFi Ingestion Pipelines:** Our platform supports scalable NiFi ingestion pipelines, enabling you to handle large volumes of data and ensure seamless integration with your existing infrastructure.
- **Tool Calling:** RAG Studio enables seamless integration with large language models (LLMs) that support function or tool calling, including OpenAI, Azure OpenAI, Amazon Bedrock, and Cloudera AI Inference service. It also provides connectivity to Model Context Protocol (MCP) tools, which can be hosted by RAG studio or used externally.

Foundation Model Integration

- **Seamless Connection:** RAG Studio seamlessly connects to proprietary foundational models or fine-tuned models (if Cloudera AI Inference service is integrated) with your organizational knowledge, empowering users to leverage the right model for every task and switch easily between them.

Using RAG Studio

Integrate with existing data infrastructure and workflows, while also maintain control over your data.

About this task

Once the RAG Studio is configured, you will be able to:

- Access the Studio Application UI, if the necessary permission is granted in the project settings
- Create chat interactions with the available large language models, where users can ask questions and get answers from the model
- Ensure that a knowledge base is used to ground answers from the large language model with direct quotes and inputs sourced from the documents in the knowledge store
- Provide feedback on each answer which will be collated in the Analytics tab and stored in ML Flow for any model evaluation or training purposes

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.

The Cloudera AI Workbenches page displays.

2. Click on the name of the workbench.

The workbenches Home page displays.

3. Click Projects, and then click New Project to create a new project.

In the left navigation pane, the new **AI Studios** option is displayed.

4. Click AI Studios and select **RAG Studio** to enter the application.

5. Start interacting with the chat function.

At this stage you can begin interacting with the RAG Studio chat function, however the chat is limited to using only knowledge from the model's training as its source, as there is no knowledge base defined yet.

- a. Write your question into the chat field at the bottom of the page, send the question and wait for the chatbot's response.
- b. Complete your question with the information you missed from the answer, or rephrase your original questions with more details added. Include more details to help the chatbot provide a more precise response.
- c. Click on the Suggested Questions drop-down list, located above the chat field and select one of the predefined questions.



Note:

To leverage RAG Studio even more, you can define a Knowledge Base as a dedicated source for your queries. Also note that switching models- under the Models tab in the top navigation bar can result in varied responses and improve results for you. For more details, see Step 6 and Step 15.

6. Click Knowledge Bases and select the Create Knowledge Base button.

The RAG Studio chat function, using the knowledge base, is a more advanced version of the chat function. It uses a knowledge base to access and retrieve information to provide more accurate and up-to-date answers to users. When you send a message, the chatbot searches the knowledge base to find relevant information and provide a response based on that information.

7. Fill in the required fields for the **Knowledge Base**.

- Name - the name of the Knowledge Base
- Chunk size - refers to the amount of data that is processed and written to the database in a single operation. Long documents are divided into smaller chunks for referencing, with the chunk size determining the size of these pieces. If unsure, it is recommended to keep the default value of 512.
- Embedding Model - The selected large language model that is used will “read” the provided documents and transform them into the vectors it needs to reference later in a process called embedding.
- Summarization model - to enable summary-based retrieval
- Advanced Options
 - Distance metrics - Cosine
 - Chunk overlap - This setting controls how much of the previous chunk's data is included in the next chunk, so as the small pieces of the larger document are referenced information at the boundaries between the pieces and are not lost

Once created and connected to a chat, the information in the knowledge base becomes the only knowledge the large language model is allowed to reference, grounding its answers in your enterprise context.

8. Fill the required **Knowledge Base** by uploading documents.

Supported file types include:

- .txt, .md, .csv
- .pdf, .docx, .pptx, .pptm, .ppt
- .jpg, .jpeg, .png
- .js
- If advanced document processing is enabled, then images and charts contained within PDFs will also be ingested

9. To begin a RAG-enabled chat, click Chats in the top-left corner. The **Chat with the LLM** field is displayed.

10. Select the Knowledge Base you would like to use for your chat, from the drop-down list, in the bottom-right corner of the Chat with the LLM field.

11. Optionally, select the Inference model to be used.

12.

Click the  icon.

The main Chat window with a chat field is displayed.

13. Configure Chat Settings if required.



You can configure the followings:

- Knowledge Base: Select the required Knowledge Base.
- Name: Provide a name for the chat.
- Response synthesizer model: Select the model that you would like to write the final answer.
- Reranking model: Select the model you want to decide what documents and snippets are the most important to reference. This feature is not available with OpenAI.
- Maximum number of documents: Select how many document chunks you want the answer to reference and incorporate. The number is set to 10 by default.
- Advanced options:
 - Enable Tool calling (Technical Preview): Enable or disable tool calling for each session and select the allowed tools using the Tools Manager. The platform automatically verifies whether the selected model supports tool calling.
 - Enable HyDE: Enable Hypothetical Document Embeddings (HyDE) during retrieval to enhance retrieval performance.
 - Enable Summary filtering: Utilize summaries to filter out retrieved chunks.
 - Disable streaming

14. Write your question into the chat field, send it and wait for a reply.

15. Check the answer you received from RAG studio.

You can also spot check RAG Studio's automatic evaluation of the answer based on the available text, the Knowledge Base:

-  : marks the level of relevancy, measures if the response and source nodes match the query. Does the question/answer pair make sense?
-  : marks faithfulness, measures if the response from a query engine matches any source nodes. Does the provided answer match the source documents well?

16.



Evaluate the answer with the help of the icons.

Anyone using the chat can optionally provide feedback on each answer, which can be used to systematically evaluate the performance of the chatbot.

17. View the evaluations, summary and analytics by selecting Analytics in the top navigation bar.

The available analytics are:

- App Metrics - It provides metrics on the overall Studio application deployed.
- Session Metrics
- Inference Metrics
- Feedback Metrics
- Auto evaluation metric averages
- Chunk relevance over time

All the underlying data is also stored in the local MLFlow instance for machine learning scientists to use as needed.

Related Information

[Launching RAG Studio within a project](#)

[Configuring RAG Studio](#)

Use cases for RAG Studio

Retrieval Augmented Generation (RAG) Studio is a powerful tool for building and deploying chatbots that use a combination of retrieval and generation to provide accurate and informative responses.

With RAG Studio, you can create chatbots that can access a knowledge base and generate responses that are tailored to the user's query. Here are some use cases for RAG chatbots across various industries:

- **Customer Service (Internal Enablement)** RAG chatbots can assist customer service agents by retrieving accurate, up-to-date information from internal knowledge bases about products, policies, or procedures. This reduces time spent searching for answers and enables faster, more consistent responses during customer interactions — without exposing the chatbot directly to end users.
- **Technical Support (Agent Assist)** Support staff can use RAG chatbots to troubleshoot technical issues more efficiently. The bot retrieves relevant documentation, FAQs, or internal repair guides, surfacing step-by-step instructions that agents can then relay to customers — improving first-contact resolution and reducing training ramp-up time.
- **Healthcare (Clinical or Admin Staff Support)** Healthcare professionals and administrative staff can reference RAG chatbots to quickly access internal guidelines, treatment protocols, or billing procedures. This helps ensure staff provide accurate information to patients while reducing the risk of compliance or documentation errors.

- Finance (Employee-Facing Advisory Support) RAG chatbots can support financial advisors, support teams, and operations staff by retrieving up-to-date guidance on financial products, compliance rules, and client onboarding processes. This ensures consistency in how internal teams advise or process customer requests — improving accuracy without giving direct chatbot access to customers.
- Travel and Hospitality (Frontline Employee Aid) Front-desk agents and travel advisors can use RAG chatbots to instantly access updated information about destinations, hotel policies, loyalty programs, or service disruptions. This enables faster and more informed responses to guest questions, helping improve satisfaction while preserving human-led interactions.
- E-commerce (Support and Sales Enablement) Internal sales and support teams can use the chatbot to quickly retrieve product specs, return policies, pricing rules, and promotional details from the knowledge base. This empowers agents to respond quickly and accurately to customer inquiries — without the chatbot being exposed to buyers directly.
- Education (Faculty and Student Support Teams) Academic advisors, admissions officers, and support staff can use the RAG chatbot to retrieve information about course offerings, degree requirements, or institutional policies. This helps them provide consistent guidance to students without relying on outdated documentation or interrupting SMEs.
- Government (Internal Public Service Enablement) RAG chatbots can help government employees and public service agents access internal guidelines, regulations, and procedural documents. This ensures more accurate and consistent communication with the public, without opening up direct AI interactions to citizens.
- Insurance: RAG chatbots can provide customers with accurate and up-to-date information about insurance policies or claims, helping to improve customer engagement and reduce the risk of insurance-related errors.

Using Fine Tuning Studio

Fine-tuning is a process that involves training a specific model to meet the standards of an organization. By using fine-tuned models, you can improve the accuracy and performance of your LLMs, reduce training time, and achieve cost benefits. Fine-Tuning Studio simplifies this process by providing a comprehensive platform for organizing data, testing prompts, training adapters, and evaluating performance.

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click Projects, and then click New Project to create a new project.
In the left navigation pane, the new **AI Studios** option is displayed.
4. Click AI Studios and select **Fine Tuning Studio**.
The Fine Tuning Studio page is displayed.
5. Select Resources in the top navigation bar for Fine Tuning Studio and click Import Base Models.
The **Import Base Models** page is displayed.
6. On the Fine Tuning Studio main page, select Resources in the top navigation bar, and click Import Datasets.
The Import Datasets page is displayed.
7. Select **Resources** in the top navigation bar for Fine Tuning Studio and click Create Prompts in the top navigation pane.
The Create prompts page is displayed.
8. Select Experiments in the top navigation bar for Fine Tuning Studio and click Train a New Adapter.
9. Fill in the required fields of the **Train a New Adapter** page with the help of the instructions provided on the page.

10. Click Start Job.
11. Select Experiments in the top navigation bar for Fine Tuning Studio and click Monitor Training Jobs.
12. Select Experiments in the top navigation bar for Fine Tuning Studio and click Run MLFlow Evaluation.



Note: This feature requires a GPU.

13. Select Experiments in the top navigation bar for Fine Tuning Studio and click View MLFlow Runs.
14. If you want to export the Fine Tuning model, go to AI Workbench in the top navigation bar for Fine Tuning Studio and click Export And Deploy Model.

You can select exporting the Fine Tuning Model to Cloudera AI Registry or to Cloudera AI Deployment. Models exported to Cloudera AI Registry can be found under Experiments in the left navigation pane, whereas models exported to Cloudera AI Model Deployment can be found under Model Deployments in the left navigation pane.

15. For an example, select Examples in the top navigation pane.
16. Export or import your metadata with the help of the Database Import Export button in the top navigation pane.

Related Information

[Launching Fine Tuning Studio within a project](#)

Fine Tuning Studio Overview

The Fine Tuning Studio is a comprehensive application and ecosystem that enables you to manage the entire lifecycle of Large Language Models (LLMs), including training, fine-tuning, and evaluation. With a streamlined approach to organizing and dispatching Cloudera AI workloads, this AI studio is specifically designed to support tasks related to LLM training and evaluation.

The Fine Tuning Studio provides a centralized platform for managing the entire process, from data preparation to model deployment, with a particular focus on Cloudera AI Jobs.

The Fine Tuning Studio supports a wide range of use cases, including:

- Ticketing Support Agent: Fine-tune LLMs to provide accurate and efficient support for customer inquiries, reducing the need for human intervention and improving customer satisfaction.
- Text to SQL conversion: Use the Fine Tuning Studio to fine-tune LLMs for text-to-SQL conversion, enabling the creation of accurate and efficient SQL queries from natural language inputs.
- Dataset detoxification: Fine-tune LLMs to detect and remove biased or toxic data from datasets, ensuring that models are trained on high-quality and diverse data.

By providing a comprehensive platform for managing the entire lifecycle of LLMs, the Fine Tuning Studio enables you to optimize your AI workloads and improve the accuracy and efficiency of your models. With its streamlined approach and focus on Cloudera AI Jobs, this AI studio is an essential tool for any organization looking to leverage the power of LLMs.



Note:

Currently Fine Tuning Studio does not support AI inference, fine tuning of huggingface models are supported.

Key Features for Fine Tuning Studio

The Fine Tuning Studio is a powerful tool that enables you to customize and optimize large language models to meet the specific needs of your organization.

With a unified workbench, you can easily import datasets, select base models, launch fine-tuning jobs, evaluate performance, and export models seamlessly.

- Unified Workbench: Import datasets, select base models, launch fine-tuning jobs, evaluate performance, and export models in a single, streamlined interface.
- Direct MLflow Integration: Track experiments and automate model registry workflows with seamless integration with MLflow.
- Flexible Interfaces: Choose from no-code or low-code visual interfaces or use the full-code Python SDK for ultimate flexibility and control.
- Domain-Specific Applications: Perfect for domain-specific applications such as healthcare, finance, and legal, where tailored accuracy is crucial.
- Easy Dataset Upload: Effortlessly upload datasets to Hugging Face, enabling broader accessibility and facilitating further model training.
- Cost-Effective: Fine-tuned smaller models are significantly more cost-effective to operate compared to large foundational models.
- Improved Performance: Fine-tuned models can outperform larger foundational models in task-specific scenarios, making them a valuable asset for organizations.

By using the Fine Tuning Studio, you can unlock the full potential of large language models and create customized models that meet the specific needs of your organization.

Using Fine Tuning Studio

Fine-tuning is a process that involves training a specific model to meet the standards of an organization. By using fine-tuned models, you can improve the accuracy and performance of your LLMs, reduce training time, and achieve cost benefits. Fine-Tuning Studio simplifies this process by providing a comprehensive platform for organizing data, testing prompts, training adapters, and evaluating performance.

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click Projects, and then click New Project to create a new project.
In the left navigation pane, the new **AI Studios** option is displayed.
4. Click AI Studios and select **Fine Tuning Studio**.
The Fine Tuning Studio page is displayed.
5. Select Resources in the top navigation bar for Fine Tuning Studio and click Import Base Models.
The **Import Base Models** page is displayed.
6. On the Fine Tuning Studio main page, select Resources in the top navigation bar, and click Import Datasets.
The Import Datasets page is displayed.
7. Select **Resources** in the top navigation bar for Fine Tuning Studio and click Create Prompts in the top navigation pane.
The Create prompts page is displayed.
8. Select Experiments in the top navigation bar for Fine Tuning Studio and click Train a New Adapter.
9. Fill in the required fields of the **Train a New Adapter** page with the help of the instructions provided on the page.
10. Click Start Job.
11. Select Experiments in the top navigation bar for Fine Tuning Studio and click Monitor Training Jobs.

12. Select Experiments in the top navigation bar for Fine Tuning Studio and click Run MLFlow Evaluation.



Note: This feature requires a GPU.

13. Select Experiments in the top navigation bar for Fine Tuning Studio and click View MLFlow Runs.
14. If you want to export the Fine Tuning model, go to AI Workbench in the top navigation bar for Fine Tuning Studio and click Export And Deploy Model.

You can select exporting the Fine Tuning Model to Cloudera AI Registry or to Cloudera AI Deployment. Models exported to Cloudera AI Registry can be found under Experiments in the left navigation pane, whereas models exported to Cloudera AI Model Deployment can be found under Model Deployments in the left navigation pane.

15. For an example, select Examples in the top navigation pane.
16. Export or import your metadata with the help of the Database Import Export button in the top navigation pane.

Related Information

[Launching Fine Tuning Studio within a project](#)

Use case for Fine Tuning Studio - Event ticketing support

To demonstrate the simplicity of building and deploying a production-ready application with Fine Tuning Studio, explore a complete example: fine-tuning a customer support agent for event ticketing.

About this task

The objective is to refine a compact, cost-efficient model capable of interpreting customer input and identifying the appropriate 'action' (such as an API call) for the downstream system to execute. The aim is to optimize a model that is lightweight enough to run on a consumer GPU while delivering accuracy comparable to that of a larger model.

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click AI Studios and select **Fine Tuning Studio**.
The Fine Tuning Studio page is displayed.
4. On the Fine Tuning Studio main page, select Resources in the top navigation bar, and click Import Datasets.
The Import Datasets page is displayed.
5. Select Import Huggingface Dataset from the tab options and import the bitext/Bitext-events-ticketing-llm-chatbot-training-dataset dataset available on Hugging Face.
This dataset consists of paired examples of customer inputs and their corresponding intent or action outputs, covering a wide range of scenarios.
6. Select Resources in the top navigation bar for Fine Tuning Studio and click Import Base Models. The **Import Base Models** page is displayed.
7. Select Import Huggingface Models from the tab options and import the bigscience/bloom-1b1 model from Hugging Face.

For details on how to import a model, see [Importing models from Hugging Face \(Technical Preview\)](#)

This model is imported to minimize the inference footprint. The goal is to train an adapter for the base model, improving its predictive performance on the specific dataset.

8. Select Resources in the top navigation bar for Fine Tuning Studio and click Create Prompts in the top navigation pane.

The Create prompts page is displayed.

9. Create the training prompt for both training and inference.

This prompt will provide the model with additional context for making accurate selections:

- a. Select Resources in the top navigation bar for Fine Tuning Studio and click Create Prompts.
- b. Name the prompt better-ticketing.
- c. Use the bitext dataset as the base for its design.
- d. Build a prompt template based on the features available in the dataset, using the Create Prompts page.
- e. Once the prompt is created, test it against the dataset to ensure it functions as expected.
- f. After verifying that everything works correctly, select the Create Prompt button to activate the prompt for use across the tool.

Here is an example of our prompt template, which leverages the instruction and intent fields from the dataset:

```
You are an event ticketing customer LLM chatbot responsible for generating a one-word, snake_case action, based on a customer input. Please provide the most relevant action based on the input from the customer below.
```

```
### CUSTOMER: {instruction}
```

```
### ACTION:
```

Completion template:

```
{intent}
```

The completed form:

Figure 1: Completed form for Create prompt action

Prompt Name
better-ticketing

Dataset
bitext/Bitext-events-ticketing-llm-chatbot-training-dataset

Dataset Columns:

- * instruction
- * intent
- * category
- * tags
- * response

Prompt Template

You are an event ticketing customer LLM chatbot responsible for generating a one-word, snake_case action, based on a customer input. Please provide the most relevant action based on the input from the customer below.

CUSTOMER: {instruction}
ACTION:

Completion Template

{intent}

Generate Prompt Example

Example Training Prompt

You are an event ticketing customer LLM chatbot responsible

CUSTOMER: I want to know more about the delivery period
ACTION: delivery_period

Example Prompt

You are an event ticketing customer LLM chatbot responsible

CUSTOMER: I want to know more about the delivery period
ACTION:

Example Completion

delivery_period

Create Prompt

10. Select Experiments in the top navigation bar for Fine Tuning Studio and click Train a New Adapter.

With the dataset, model, and prompt selected, you need to train a new adapter for the bloom-1b1 model to improve its ability to handle customer requests accurately.

Fill out the required fields, that is, the name of the adapter, the dataset for training, and the training prompt to be used.

For this example, two L40S GPUs were available for training, so the Multi-Node training type was selected. The model was trained for 2 epochs using 90% of the dataset, with the remaining 10% reserved for evaluation and testing.

11. Select Experiments in the top navigation bar for Fine Tuning Studio and click Monitor Training jobs.

It allows you to track the status of the training job and access a deep link to the Cloudera AI Job for viewing log outputs. Using two L40S GPUs, the training on 2 epochs of the `bitext` dataset was completed in 10 minutes.

12. Select Experiments in the top navigation bar for Fine Tuning Studio and click Local Adapter Comparison to check the performance of the Adapter.

After the training job is complete, it is important to "spot check" the adapter's performance to ensure it was trained successfully.

For example, consider a simple customer input taken directly from the bitext dataset: *I have to get a refund, I need assistance.* where the desired output action is `get_refund`. Comparing the output of the base model with that of the trained adapter clearly demonstrates that the training process significantly improved the adapter's performance.

Figure 2: Inference Results

Inference Results

Base Model: bigscience/bloom-1b1

```
ı want to see if it is possible to give me refunds.  
##          * The client wants to cancel their reservation c  
### - ı can tell him that he has to pay or not  
### - ı can send back a letter requesting
```

Adapter: bloom-1b-ticketing

```
get_refund
```

13. Select Experiments in the top navigation bar for Fine Tuning Studio and click Run MLFlow Evaluation.

You can evaluate the performance against the “test” portion of the dataset. For this example, the performance of 1) the bigscience/bloom-1b1 base model 2) the same base model with our newly trained better-ticketing adapter activated, and finally 3) a larger mistral-7b-instruct model are compared:

Dataset used : bitext/Bitext-events-ticketing-llm-chatbot-training-dataset

Aggregated Results

metric	bigscience/bloom-1b1	bigscience/bloom-1b1 + better-ticketing-adapter	unsloth/mistral-7b-instruct-v0.3
latency/mean	3.3002	0.0689	1.4363
toxicity/v1/mean	0.0118	0.0007	0.0022
flesch_kincaid_grade_level/v1/mean	10.5024	35.5129	23.0669
ari_grade_level/v1/mean	17.9919	58.1839	33.625
rouge1/v1/mean	0.015	0.9847	0.1436
rougeLsum/v1/mean	0.015	0.9847	0.1397
rougeL/v1/mean	0.0145	0.9847	0.1395
exact_match/v1	0	0	0

As demonstrated, the rougeL metric (a more complex variation of exact match) for the 1B model adapter is significantly higher than the same metric for an untrained 7B model. This highlights how successfully an adapter for a smaller, cost-effective model has been trained, which outperforms a much larger model.

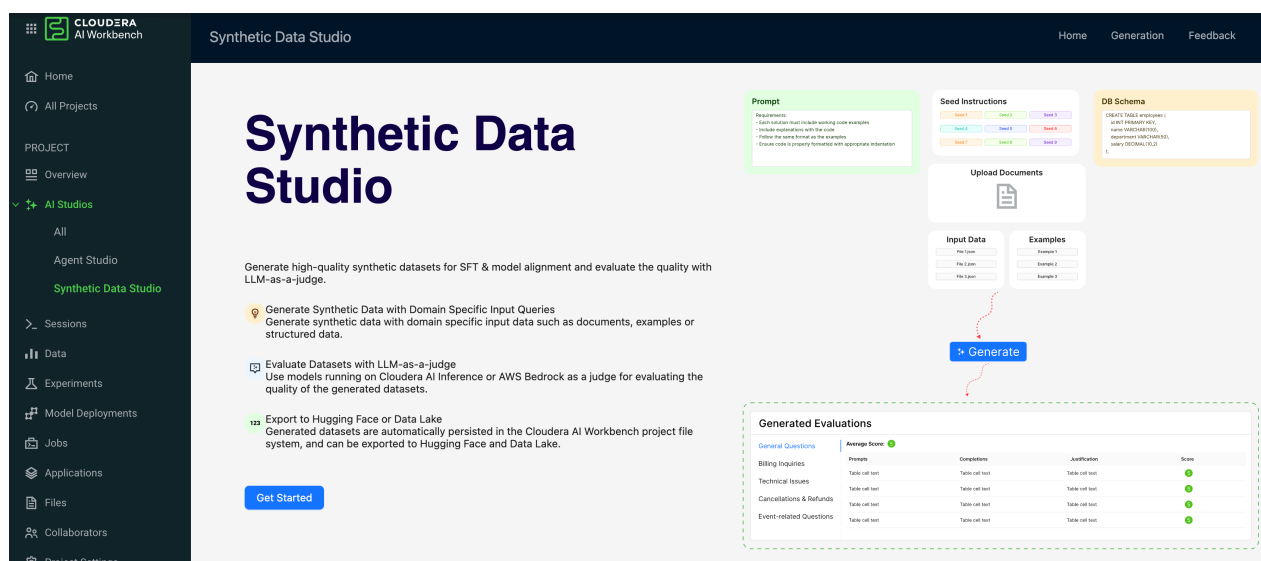
While the larger 7B model may excel at generalized tasks, it lacks fine-tuning on the specific 'actions' the model can take based on customer input. As a result, the non-fine-tuned 7B model would not perform as effectively as the fine-tuned 1B model in a production environment.

Using Synthetic Data Studio

Synthetic Data Studio is a Cloudera AI Platform application designed to enable scalable synthetic data generation, helping enterprises tackle complex AI use cases through supervised fine-tuning, model alignment, knowledge distillation, and custom data curation.

Synthetic Data Studio Overview

Synthetic Data Studio (SDS) is an application on the Cloudera AI Platform designed to enable scalable synthetic data generation. It empowers enterprises to address complex AI use cases by supporting supervised fine-tuning, model alignment, knowledge distillation, and custom data curation.



Synthetic Data Studio use cases

Explore key use cases where Synthetic Data Studio can help drive innovation, such as model fine-tuning, knowledge distillation, and custom data curation, empowering businesses to build smarter, more efficient AI solutions..

Use Case 1: [Enhancing Customer Support LLMs with Privacy-Compliant Training](#):

Synthetic Data Studio (SDS) addresses the challenge of distilling knowledge from frontier large language models (LLMs) while adhering to strict data privacy regulations. By generating synthetic customer support interactions and analytics, SDS enables the distillation of cloud-based models and fine-tuning of smaller, faster models like Meta-Llama-3.1-8B-Instruct, achieving a 70% win rate over a baseline Goliath-120B model in real-world evaluations without exposing customer data. This approach ensures high-quality training without exposing sensitive customer data.

Use Case 2: [Generating Structured Data adhering to the original dataset statistical characteristics](#).

SDS solves the problem of approximating the statistical properties of real-world datasets for analytics and modeling, particularly in scenarios where raw data is restricted or insufficient. Using clustering and seed instructions, SDS creates synthetic tabular data (e.g., financial sensitive data) that preserves distributions, correlations, and business rules. This ensures synthetic data aligns with original datasets in metrics like mean, standard deviation, and KL divergence, enabling privacy-compliant analysis and model training.

Use Case 3: [Accelerating LLM Training for Coding Tasks with Synthetic Code](#)

SDS curates large coding datasets to improve code generation LLMs. It synthesizes code questions, solutions, and unit tests for automated testing. Fine-tuning coding models with such data reduces coding generation errors.

Use Case 4: [Scaling Evaluation of LLMs, RAG Systems, and Agents](#)

Manual evaluation of LLMs for various tasks such as coding is time-intensive. SDS automates this process by generating synthetic tasks, tests, and using LLM-as-a-judge prompts to validate outputs at scale. SDS also allows humans to filter out bad samples by inspecting or testing the generated data.

Key Features of Synthetic Data Studio

Learn the key features of Synthetic Data Studio and how they enable scalable synthetic data generation for advanced AI use cases.

- Supervised Fine-Tuning Dataset Generation
 - Automatically generates high-quality prompt-completion pairs from raw or redacted documents by creating question-answer sets or summaries.
 - Ideal for fine-tuning machine learning models, especially for unstructured content or when data availability is limited.
 - Supports the creation of task-specific datasets tailored to enterprise needs.
- Document-Based Generation
 - Advanced document processing capabilities allow users to generate synthetic data directly from uploaded document collections.
 - Ensures the creation of domain-specific datasets that are consistent with existing documentation and knowledge bases.
- Custom Generation Workflows
 - Offers a flexible workflow system to process user-provided inputs and create tailored data generation pipelines.
 - Guarantees that the generated data aligns perfectly with specific use cases and enterprise requirements.
- Evaluation Workflow
 - Supports the evaluation of generated supervised fine-tuning (SFT) datasets using an LLM-as-a-judge.
 - Provides evaluation scores and detailed justifications for each generated prompt-completion pair.
 - Allows you to define custom scoring criteria and justification parameters for precise quality assessments.
 - Ensures datasets meet the highest quality standards before being used for fine-tuning.
- Export Functionality
 - Enables datasets to be exported to the Project File System for easy access.
 - Supports seamless uploading of datasets to Hugging Face for broader accessibility and further model training.

Launching Synthetic Data Studio within a project

You can launch Synthetic Data Studio on the Cloudera AI Platform to generate datasets and evaluate them.

Before you begin

Agent Studio integrates with two major enterprise inference services:

- Cloudera AI Inference Service: It offers enterprise-grade deployment options.

To enable Cloudera AI Inference service for Synthetic Data Studio, ensure the followings:

- The environment variable responsible for enabling Cloudera AI Inference service is `CDP_TOKEN`. By default `CDP_TOKEN` is set to null. If left as null, the application will use the JWT stored at `/tmp/jwt` to run Cloudera AI Inference service. Alternatively, if you provide a value for `CDP_TOKEN` during the pre-installation configuration of environment variables, it will override the default and be used for authentication.
- Ensure that the Cloudera AI Inference service endpoints and model IDs are readily available. You will be prompted to provide these details if you choose Cloudera AI Inference service as the AI inference option in Synthetic Data Studio (SDS).
- All endpoints used must conform to the OpenAI API standard.
- For Cloudera AI on premises, you must use `CDP_TOKEN` for authentication. Auto-generated tokens stored in `/tmp/jwt/` are not yet available in the Cloudera AI on premises version.

For more details, see [Authenticating Cloudera AI Inference service](#).

- AWS Bedrock: It provides scalable cloud-based inference.



Note: To use AWS Bedrock as the inference service for Synthetic Data Studio, an active internet connection is required. Alternatively, in an air-gapped environment, the relevant endpoints must be whitelisted.

Environment Variables: Before installation, Synthetic Data Studio must be configured with the necessary environment variables - CDP_TOKEN- to enable the Cloudera AI Inference service.

- AWS_DEFAULT_REGION: Defaults to the us-east-1 region.*
- AWS_ACCESS_KEY_ID: Your AWS access key ID.*
- AWS_SECRET_ACCESS_KEY: Your AWS secret access key.*
- Hf_token: Your Hugging Face token for exporting datasets.
- Hf_username: Your Hugging Face username.
- CDP_TOKEN: Overrides the JWT token for Cloudera AI Inference service.



Note: The AWS environment variables to function properly either an active Internet connection is required or alternatively, in an air-gapped environment, the relevant endpoints must be whitelisted.

Host names: For air-gapped installations that use a proxy setup, it is essential to whitelist the necessary URLs in your firewall rules. For a list of hostnames to whitelist, see [Host names and endpoints required for AI Studios](#).

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click Projects, and then click New Project to create a new project.
In the left navigation pane, the new **AI Studios** option is displayed.
4. Click AI Studios.
5. Click the Launch button in the **Synthetic Data Studio** box. The **Configure Studio: Synthetic Data Studio** page is displayed.
6. Set the environment variables for the Synthetic Data Studio, using the details mentioned in the prerequisites.
7. Select the Runtime version.
8. Click Launch AI Studio.
The **Synthetic Data Studio** page is displayed.
After launching, you can view the list of tasks being executed as part of the AI studio deployment.
9. After configuration, **Synthetic Data Studio** is displayed in the left navigation page under AI Studios.
10. Click **Synthetic Data Studio** and click **Get Started**.
You can generate synthetic datasets for training models and evaluate the generated datasets for fine-tuning LLMs on this page.

Generating synthetic data for fine-tuning models

You can either choose from pre-defined templates or create your own. The pre-defined Code Generation and Text-to-SQL templates offer curated prompts, seeds (explained below), and examples to generate datasets. Alternatively, the Custom template lets you define every aspect from scratch, enabling the creation of synthetic datasets tailored specifically to your enterprise use cases.

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.

3. Click AI Studios.
4. [Launch the Synthetic Data Studio](#).
5. Under AI Studios, click Synthetic Data Studio, and then click Get Started.
6. Under Create Datasets, click Getting Started. The Synthetic Dataset Studio page is displayed.

7. In the Configure tab, specify the following:

- a. In Dataset Display Name, provide a name for your dataset.
- b. In Model Provider, select AWS Bedrock or Cloudera AI Inference as the model provider.



If Cloudera AI Inference is selected, specify the Model ID and Cloudera AI Inference Endpoint. You can obtain the Model ID from the Model Endpoint Details page, and you can select a target model such as, LLaMA, Mistral, and so on. If Model endpoints are in the same cluster where the application is hosted, a JWT token is sufficient. Otherwise, a CDP token must be specified in the environment variables.

- c. In Workflow, choose the tuning type:

1. Supervised Fine-Tuning: Generate prompt-completion pairs with or without additional documents. For example, PDFs, DOCs, TXTs, and so on.
2. Custom Data Generation: Use a user-uploaded JSON array to generate responses. This workflow allows you to provide your inputs and instructions and generate tailored outputs for the corresponding inputs.

For Custom Data Generation, define the following to generate the final pair of responses:

- a. Input Key: Specify the Key or Column name within the uploaded JSOM whose values will be used for generation.
- b. Output Key: Define the key name in the final output. If left empty, it defaults to Prompt.
- c. Output Value: Define the name of the generated series values corresponding to the input key. If left empty, it defaults to Completion.
- d. In Files, select input files from your project file system for the chosen workflow.

8. In the Prompt tab, specify the following:
 - a. In Prompt, create a custom prompt manually, select from predefined templates, or allow the LLM to generate a prompt based on your use case description.
 - b. In Seeds Instructions, provide seed instructions to guide the LLM in diversifying the dataset. Seeds act as prompts that influence the style, domain, or subject matter of the output.
Examples for Code Generation
 - Algorithms for Operation Research
 - Web Development with Flask
 - PyTorch for Reinforcement Learning
 Example Seeds for language translation:
 - Poems
 - Greetings in Formal Communication
 - Haikus
 - c. In Entries Per Seed, specify the number of entries to generate for each seed defined in Seeds Instructions.
 - d. Under Parameters, adjust the model parameters for Temperature, Top K, and Top P.
 1. Temperature: Low temperature improves accuracy, while high temperature enhances diversity.
 2. Top K: Increases the search space and response options for the LLM but may impact response speed.
 3. Top P: Controls diversity and randomness in the generated tokens. Higher values increase diversity and randomness, while lower values reduce both.
9. In the Examples tab, view the details of example prompts.
 - a. Under Actions, click  to modify an example prompt or response.
 - b. Under Actions, click  to remove the example prompt.
 - c. Click Add Example to create new example prompts.
10. In the Summary tab, review your settings. Click Generate to generate the dataset, or click Previous to return to earlier tabs and make changes.
11. In the Finish tab, view the status of the dataset creation process. The generated Prompts and Completions will be displayed. The output will be saved in the Project File System within the Cloudera environment.


Evaluating generated datasets for fine-tuning LLMs

Learn how to evaluate synthetic data generated in previous steps using the Large Language Model as the evaluator.

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click AI Studios.
4. Under AI Studios, click Synthetic Data Studio.
5. Under Evaluate, click Getting Started. The Evaluator page is displayed.
6. In Display Name, provide a name for the evaluation result.
7. In Prompt, enter your custom prompt, or click Restore Default Prompt to reset to the default prompt.
8. In Model Provider, select either AWS Bedrock or Cloudera AI Inference as a model provider.
9. In Model, select your desired model for the evaluation.

10. In Evaluation Examples, update the justification text and score provided by the LLM as needed.
11. Under Parameters, modify the model parameters for Temperature, Top K, and Top P to customize the model's behavior.
12. Click Evaluate to start the evaluation.

After completing the evaluation, you can access the generated evaluation report. Click  on the Actions column to re-evaluate the dataset, view the evaluation results, or remove the evaluation report.

Managing generated datasets

You can manage datasets by performing actions such as viewing, previewing, generating, evaluating, exporting, or removing them based on your requirements.


- View Dataset Details: Open the dataset file within your project to examine its contents.
- View in Preview: Preview the dataset before performing further actions to ensure it meets your needs.
- Generate Dataset: Regenerate the dataset with updated parameters, such as using a different model or configuration.
- Evaluate Dataset: Assess the quality and suitability of an existing dataset for your specific use case.
- Export Dataset: Export the dataset to the Project File System or the Hugging Face website for broader accessibility and additional model training.
- Remove Dataset: Delete the dataset from your project to free up storage or eliminate unnecessary data.

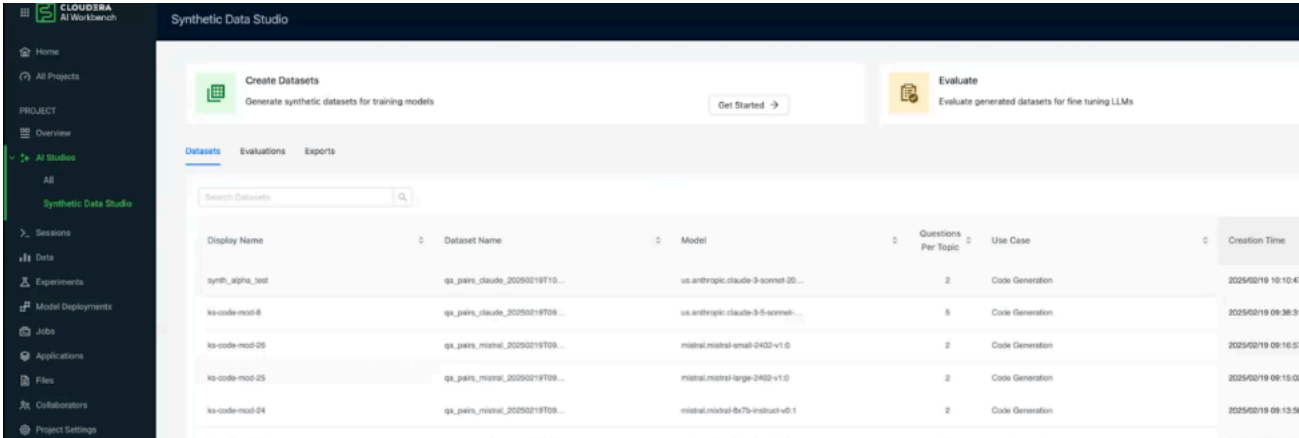
1. In the Cloudera console, click the **Cloudera AI** tile.

The Cloudera AI Workbenches page displays.

2. Click on the name of the workbench.

The workbenches Home page displays.

3. Click AI Studios.
4. Click AI Studios, click Synthetic Data Studio.
5. Locate the dataset you want to manage and click  on the Actions column next to it.
6. Choose the desired action, for example, View, Preview, Generate, Evaluate, Export, or Remove, to manage the



dataset.

Use case: Data generation for ticketing system using Synthetic Data Studio

This example illustrates how to generate a synthetic dataset for an agent ticketing use case.

Scenario Overview:

In this scenario, a user interacts with a customer support system by asking questions that require the system to direct the user to the appropriate resource. However, due to the absence of readily available training data or privacy restrictions on customer data, synthetic data is created to simulate both user questions and system responses.

Objective:

The synthetic data generation process enables knowledge distillation from larger proprietary datasets while addressing privacy concerns. This approach allows a smaller language model (SLM) to be fine-tuned to respond effectively, without relying on sensitive customer data.

Generating synthetic data for a ticketing use case using the Supervised Fine-Tuning workflow

Follow the steps below to generate synthetic data for a ticketing use case using the Supervised Fine-Tuning workflow in Synthetic Data Studio.

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click AI Studios.
4. [Launch the Synthetic Data Studio..](#)
5. Under AI Studios, click Synthetic Data Studio, and then click Get Started.
6. Under Create Datasets, click Getting Started. The Synthetic Dataset Studio page is displayed.

7. In the Configure tab, specify the following:
 - a. In Dataset Display Name, enter Ticketing dataset.
 - b. In Model Provider, select AWS Bedrock.
 - c. In Model ID, enter us.anthropic.claude-3-5-sonnet-20241022-v2:0 .
 - d. In Workflow, select Supervised Fine-Tuning.
 - e. In Template, select Custom.

8. In the Prompt tab, specify the following:

- a. In Prompt, write a prompt that instructs the LLM to create both user queries and system responses. The workflow generates both the user prompt and the completion in a single step.

Example prompt: The following example instructs the LLM by giving general guidelines on creating a prompt and a completion. Then, give a list of requirements for the data, such as the use of respectful language, the level of detail, and so on. Finally, explains the possible choices of completion and how the system will create the user prompts (queries) along with the system completions (system response).

```
Generate authentic customer support ticket interactions with a user query and system response.
For each user query, the system generates a keyword used to forward the user to the appropriate subsystem.
Requirements for user queries:
- Use professional, respectful language.
- Avoid assumptions about demographics or identity.
- Include realistic technical details when relevant.
- Provide clear, actionable solutions.
- Use inclusive terminology.
- Maintain a helpful, solution-focused tone.
- Include relevant troubleshooting steps where applicable.
- Follow standard customer service best practices.

Each response should be a single id from the following list:
cancel_ticket, customer_service, pay, report_payment_issue
Here are the explanations of the responses:
cancel_ticket means that the customer wants to cancel the ticket.
customer_service means that customer wants to talk to customer service.
pay means that the customer wants to pay the bill.
report_payment_issue means that the customer is facing payment issues and wants to be forwarded to the billing department to resolve the issue.
```

- b. In Seeds Instructions, define seed topics to diversify the generated dataset.

```
Cancellation & Refunds
Event inquiries
Billing Inquires
```


General Inquires

- c. In Entries Per Seed, specify 5 as the number of entries to generate for each seed defined in Seeds Instructions.
- d. Under Parameters, adjust the following model parameters:
 - 1. Temperature: Set to 1.0 to allow the LLM to generate diverse synthetic data.
 - 2. Top K: Set to 100 to explore a wide range of possible solutions.
 - 3. Top P: Set to 1.0 for broader exploration of outputs.
 - 4. Max Tokens. Set to 2048 or adjust based on the size of the generated text. For problems with larger generated text, consider increasing Max Tokens.
- 9. In the Examples tab, view the details of example prompts. Using examples, you can teach the LLM how to structure the prompt and completions of the generated data.
 - a. Under Actions, click Add Example and define prompts and completions so that the LLM knows the format of the data to be generated.

Click Add Example to add the following prompts:

Table 1:

Field name	Value
Example 1 - Prompt	I have received a message that I owe \$300 and I was instructed to pay the bill online. I already paid this amount and I am wondering why I received this message.
Example 1 - Completion	report_payment_issue
Example 2 - Prompt	I have received two payment invoices and need to pay my bills using a credit card.
Example 2 - Completion	pay
Example 3 - Prompt	I will not be able to attend the presentation and would like to cancel my RSVP.
Example 3 - Completion	cancel_ticket
Example 4 - Prompt	I am having questions regarding the exact time, location, and requirements of the event and would like to talk to customer service.
Example 4 - Completion	customer_service

- 10. In the Summary tab, review all the data generation parameters to confirm that everything is as expected. Click Generate to initiate dataset generation. Alternatively, click Previous to return to previous tabs and make any necessary changes.
- 11. In the Finish tab, view the status of the dataset creation. The generated Prompts and Completions will be displayed. The output dataset will be saved in the Project File System within the Cloudera environment.

Results

This workflow ensures that high-quality synthetic datasets are generated for fine-tuning LLMs to handle customer support ticketing use cases effectively.


Evaluating the generated dataset

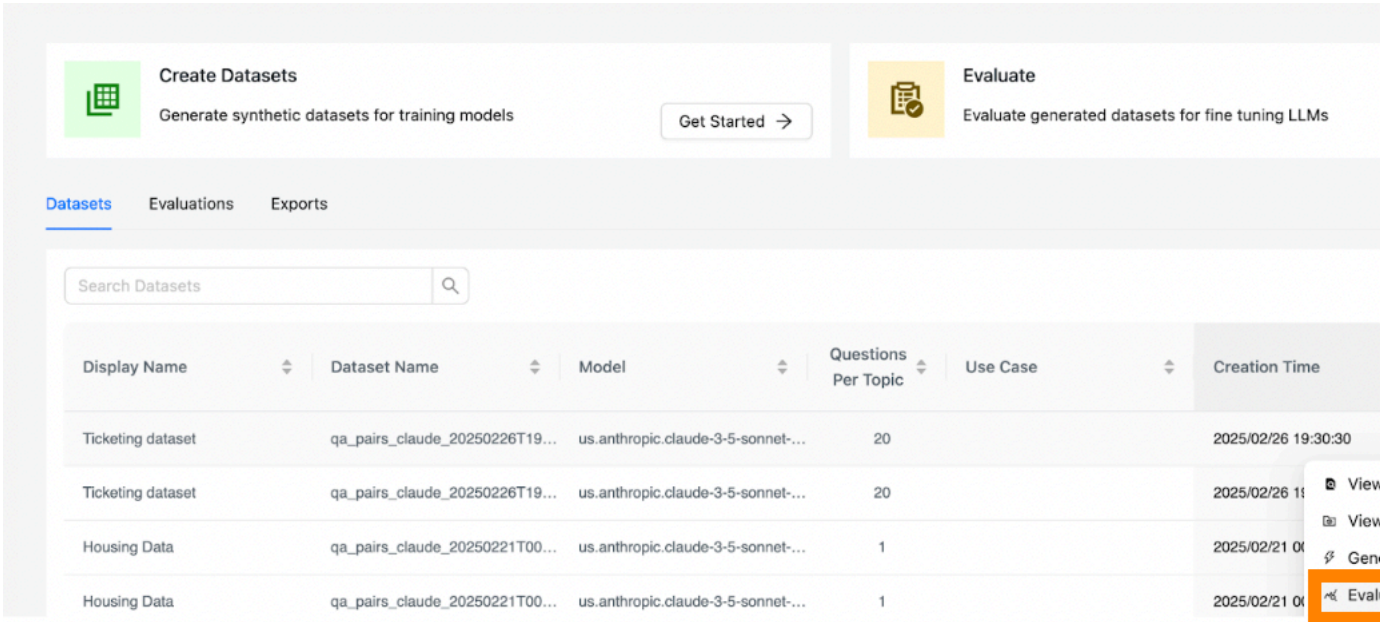
After generating a dataset, it is essential to evaluate its quality to ensure that only the highest-quality data is retained. This can be achieved using the LLM-as-a-judge approach, which evaluates and scores prompts and completions, filtering out irrelevant or low-quality data.

Procedure

- 1. In the Cloudera console, click the **Cloudera AI** tile.

The Cloudera AI Workbenches page displays.

- 2. Click on the name of the workbench.
The workbenches Home page displays.
- 3. Click AI Studios.
- 4. In the Synthetic Data Studio page, locate the dataset you want to evaluate.
- 5. Click  next to the dataset and click Evaluate Dataset.



- 6. Define a prompt to guide the LLM-as-a-judge on how to evaluate and score the dataset. Example prompt for evaluation:

Table 2:

Field Name	Value
Evaluation Display name	Ticketing Dataset Evaluation

Field Name	Value
Prompt	<p>You are given a user query for a ticketing support system and the system responses which is a keyword that is used to forward the user to the specific subsystem.</p> <p>Evaluate whether the queries:</p> <ul style="list-style-type: none"> - Use professional, respectful language - Avoid assumptions about demographics or identity - Provide enough details to solve the problem <p>Evaluate whether the responses use only one of the the four following keywords: cancel_ticket,customer_service,pay,report_payment_issue</p> <p>Evaluate whether the solutions and responses are correctly matched based on the following definitions:</p> <p>cancel_ticket means that the customer wants to cancel the ticket.</p> <p>customer_service means that customer wants to talk to customer service.</p> <p>pay means that the customer wants to pay the bill.</p> <p>report_payment_issue means that the customer is facing payment issues and wants to be forwarded to the billing department to resolve the issue.</p> <p>Give a score of 1-5 based on the following instructions:</p> <p>If the responses don't match the four keywords give always value 1.</p> <p>Rate the quality of the queries and responses based on the instructions give a rating between 1 to 5.</p>
Entries per seed	5
Temperature	0
TopK	100
Max Tokens	2048

- After defining the prompt and parameters, click Evaluate to begin the evaluation process.
- Once the evaluation is complete, select the evaluation and click Preview to review the evaluated dataset. Each sample in the dataset will include fields for scoring and justification.
- Understand the evaluation output by reviewing the Justification and Score fields. The Justification field explains how the LLM scored each query and completion. The Score field is a numerical value (1–5) that can be used to filter data based on quality.
- Click Download to download the evaluated dataset for further analysis or use the dataset for additional processing or fine-tuning of your language model.

Results

This evaluation process helps ensure that the generated dataset meets quality standards, providing a strong foundation for subsequent fine-tuning and training tasks.

Using Agent Studio

Cloudera Agent Studio is a versatile low-code to high-code platform for building, testing, and deploying multi-agent workflows.

Agent Studio Overview

Cloudera Agent Studio is a versatile low-code to high-code platform for building, testing, and deploying multi-agent workflows. It offers an intuitive interface that allows users to create AI agents, assign tasks, custom AI tools and combine them into advanced automated workflows—with little to no coding required. This empowers broader teams, including business users, to prototype and experiment quickly. For more advanced use cases, you can seamlessly switch to high-code mode using Cloudera AI Workbench to build and customize agents and tools from scratch.

Agent Studio supports full lifecycle management of production-ready agent workflows, complete with built-in observability and logging for monitoring and troubleshooting.

Designed for a wide range of enterprise use cases, Agent Studio can orchestrate workflows that collect real-time data, connect to various data sources, process and analyze information, and deliver outputs such as trend analysis, reports, copilots, and personalized recommendations. It enables the creation of intelligent agentic applications that support both insights generation and decision-making.

Key Features of Agent Studio

Learn the key features of Agent Studio and how they empower users to build and manage intelligent agents effectively.

- **Design Agentic Workflows with a Low-Code Interface:** Agent Studio leverages workflows as the foundational structure for AI agents. A workflow represents a network of collaborative agents working together to perform a sequence of interconnected tasks. With the low-code interface, you can easily:
 - Create agents and assign them specific tasks and tools.
 - Define agentic workflows as either conversational or task-oriented.
 - Assign a manager agent to oversee and coordinate interactions between agents.
- **Pre-Built and Custom Workflow Templates:** Agent Studio includes a comprehensive library of pre-built workflows and tools to expedite development. Additionally, you can also create custom workflows and tools, which can be saved and managed as reusable templates for team-wide use.
- **Agent and Task Configuration:** Agents are intelligent entities capable of decision-making, executing actions, and collaboration. With Agent Studio, you can:
 - Define each agent's role, goal, and contextual backstory.
 - Assign specific tasks and tools to individual agents.
 - Use AI-assisted authoring to generate agents using natural language input.
- **Development and Extension of Custom Tools:** Tools allow agents to interact with external systems, such as APIs, databases, or business logic, beyond the scope of LLMs. Agent Studio supports:
 - Use of built-in tools or development of custom tools.
 - Extension of existing tools by modifying logic, parameters, or API integrations.
 - Development of new tools from scratch using Cloudera AI Workbench notebooks and Python sessions.
- **Integration with AI Models:** Easily integrate your workflow with the LLM of your choice by registering a model and providing API key information. You can connect to Cloudera AI Inferencing or use any OpenAI-compatible model provider.
- **Test and Debug Workflows:** Ensure workflow quality and correctness through comprehensive testing and debugging features:
 - Execute workflows with test inputs and visualize their execution using interactive flow diagrams.
 - Access step-by-step logs and animations in real-time.
 - Use playback mode to analyze and debug specific workflow steps.
- **Deploy Workflows as Endpoints and Applications:** Once a workflow is finalized, workflow can be deployed for production use. Deployment capabilities include
 - Automatic generation of a Workbench Model endpoint.
 - Creation of a default user-facing application for interaction with the workflow.

- **Monitor Workflow Performance:** Agent Studio integrates with Phoenix, an open-source observability tool, to provide:
 - Real-time monitoring of latency, token usage, inputs, outputs, and execution traces.
 - Access to a dedicated observability panel within the Workflow UI and Phoenix dashboards in the Cloudera AI Workbench.
- **User-Facing Applications and Custom UI Development:** Each deployed workflow includes a default front-end interface for users. To further customize the experience, build your own applications using Agent Studio's Python SDK and APIs.

Use Cases of Agent Studio

Explore key use cases where organizations can leverage Agent Studio to build and orchestrate complex, goal-oriented AI systems that automate multi-step workflows.

- **Agentic DevOps:** AI agents facilitate infrastructure by performing key tasks such as analyzing logs, handling incidents, summarizing insights, and automating code integrations across clusters. This enables efficient and reliable DevOps operations with minimal human intervention.
- **Conversational Workflow Automation:** Conversational AI agents automate complex workflows, including tasks such as market analysis, portfolio monitoring, and providing personalized investment advice. These agents leverage natural language processing to interact with users and execute sophisticated processes seamlessly.
- **Automated Report Generation:** AI agents gather data from multiple systems, validate numerical accuracy, and generate concise, summarized reports. This automation significantly accelerates decision-making processes by providing business users with reliable, actionable insights.
- **Intelligent Data Processing:** AI agents are utilized for intelligent handling of data from diverse sources. They can read, classify, extract, and route information from structured and unstructured documents, such as contracts or invoices, while offering business users an interactive interface for streamlined data processing.
- **KYC Automation:** AI agents collaborate to automate Know Your Customer (KYC) processes. They verify customer identities, screen individuals against regulatory watchlists, and assess risk factors, thereby streamlining onboarding workflows and ensuring compliance with regulatory requirements.

Launching Agent Studio within a Project

Agent Studio is compatible with Cloudera AI Inference service and enabling you to build, test, and deploy multi-agent workflows.

Before you begin

Agent Studio integrates with the following major enterprise inference services:

- Cloudera AI Inference Service
- Azure OpenAI
- OpenAI
- OpenAI compatible models
- AWS Bedrock

Host names: For air-gapped installations that use a proxy setup, it is essential to whitelist the necessary URLs in your firewall rules. For a list of hostnames to whitelist, see [Host names and endpoints required for AI Studios](#).

Procedure

1. In the Cloudera console, click the **Cloudera AI** tile.

The Cloudera AI Workbenches page displays.

2. Click on the name of the workbench.

The workbenches Home page displays.

3. Click Projects, and then click New Project to create a new project.

In the left navigation pane, the new **AI Studios** option is displayed.

4. Click AI Studios.

5. Click the Launch button in the **Agent Studio** box. The **Configure Studio: Agent Studio** page is displayed.

Configure Studio: Agent Studio

Studio name: Agent Studio

Cloudera AI Agent Studio is a platform for building, testing, and deploying AI agents and workflows. It provides an intuitive interface for creating custom tools, agents, and combining them into sophisticated automated workflows.

- Agents
- Agentic Workflows
- GenAI

ⓘ

IMPORTANT: Please read the following before proceeding.

This AMP includes or otherwise depends on certain third party software packages. Information about such third party software packages are made available in the notice file associated with this AMP. By configuring and la...

▼ More Details

Environment Variables

The settings below were defined by the AMP:

Name	Value	Description
* AGENT_STUDIO_NUM_WORKFLOW_RUNNERS	<div>5</div>	Number of workflow runners to spawn for testing workflows within Agent Studio. If multiple concurrent users of Agent Studio are expected, you can increase this number accordingly.

Runtime

Editor ⓘ

Kernel ⓘ

Edition ⓘ

Version

PBJ Workbench

Python 3.10

Standard

2025.01

Runtime Image
- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-pbj-workbench-python3.10-standard:2025.01.3-b8

ⓘ

No Runtime Addon is required for this AMP.

Setup Steps

- ☒ Execute AMP setup steps

6. Set the environment variables for the Agent Studio.
7. Select the Runtime version.
8. Click Launch AI Studio.

The **AI Studio Setup Steps** page is displayed.

After launching, you can view the list of tasks being executed as part of the AI studio deployment.

9. After configuration, **Agent Studio** is displayed in the left navigation page under AI Studios.
10. Click **Agent Studio** and click **Get Started** to orchestrate AI agents to collaborate on complex tasks, powered by custom tools and seamless workflow automation.

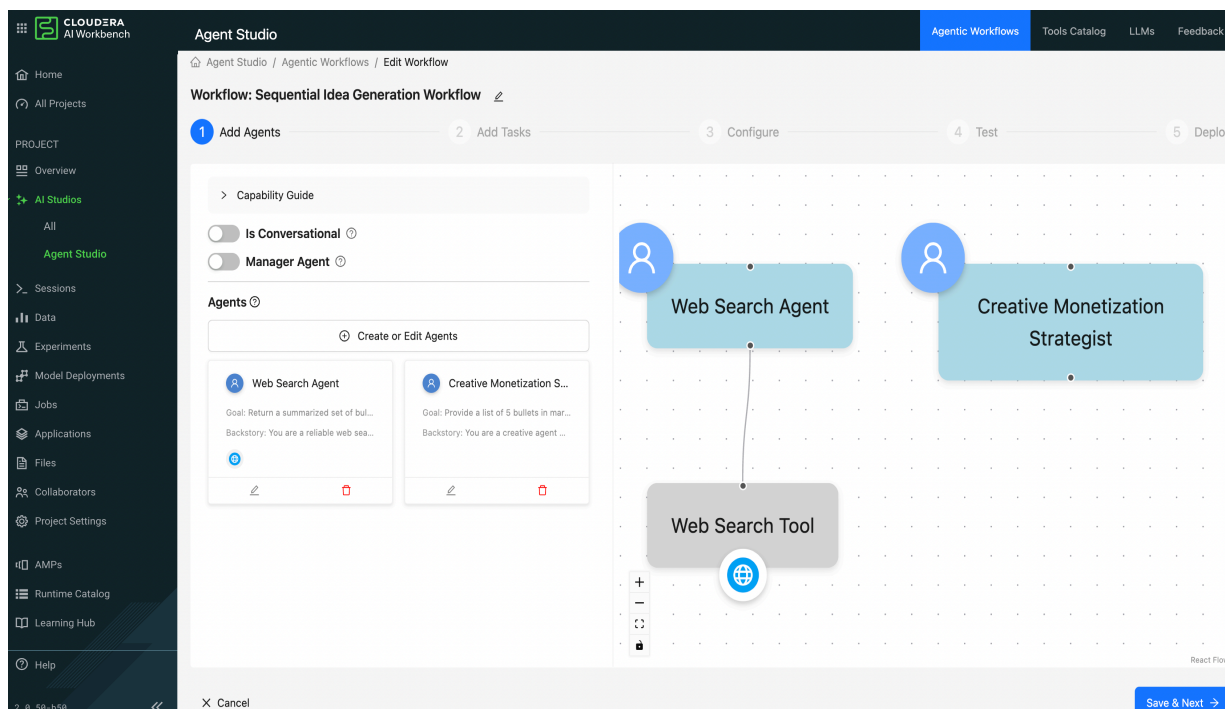
Use case: Sequential Idea Generation Workflow

This example illustrates

Procedure

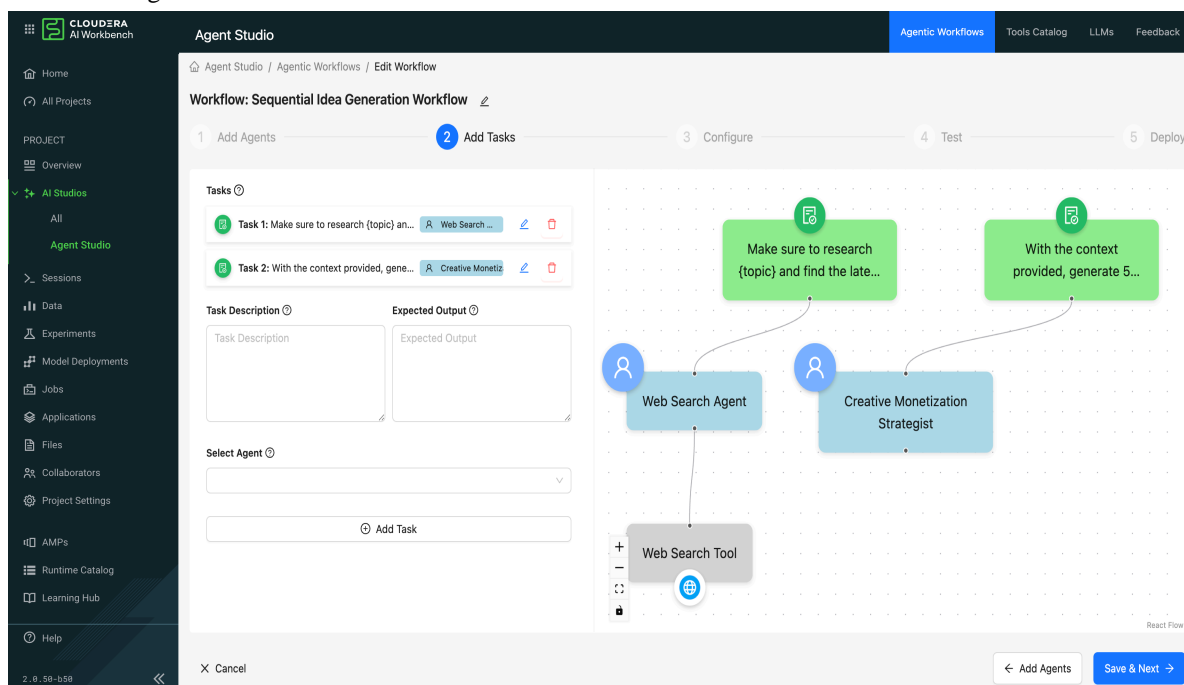
1. In the Cloudera console, click the **Cloudera AI** tile.
The Cloudera AI Workbenches page displays.
2. Click on the name of the workbench.
The workbenches Home page displays.
3. Click Projects, and then click New Project to create a new project.
In the left navigation pane, the new **AI Studios** option is displayed.
4. Click AI Studios.
5. Launch the Agent Studio within a project.
6. Click Create in the Create Agentic Workflow block to create a new workflow.
The Create New Workflow page displays.
7. In Create New Workflow page, in Workflow Name, provide Sequential Idea Generation Workflow as the name for your workflow and click Create Workflow.

8. In Add Agents page, configure the following:



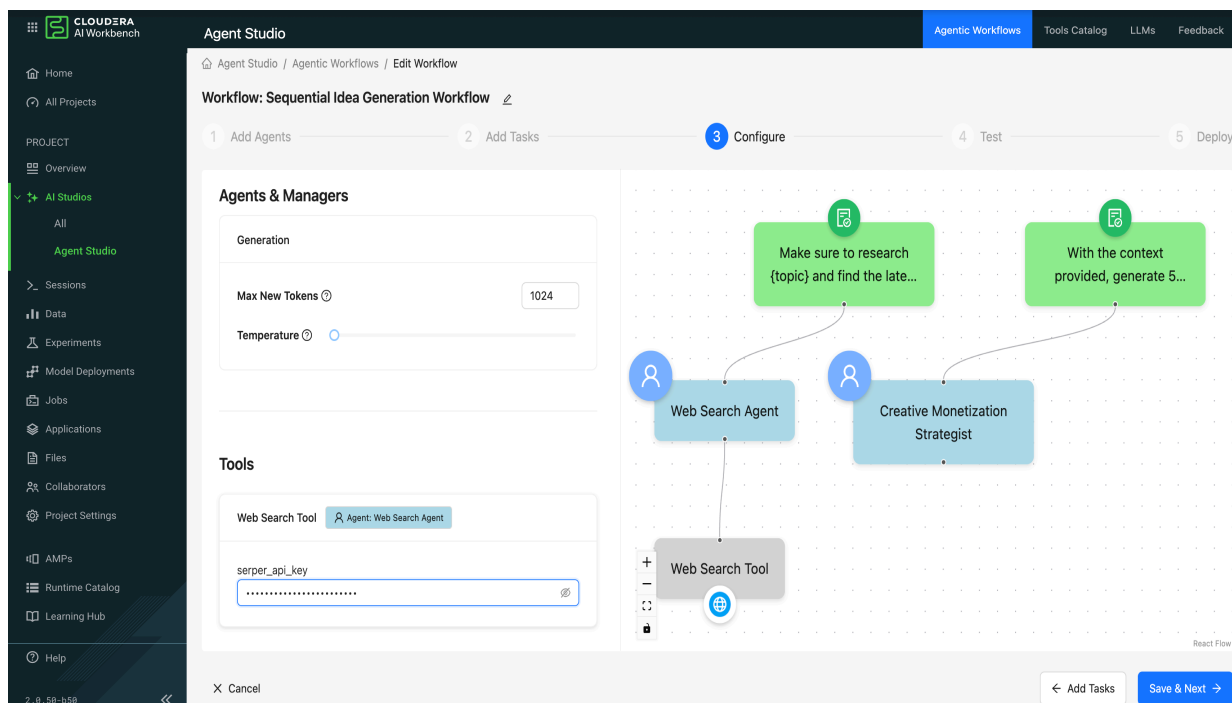
- a. Under Capabilities, turn off both the Is Conversational and Manager Agent toggle buttons.
- b. Under Agents, click Create or Edit Agents. The Create or Edit Agent window displays.
 1. Create the first Agent with the following details:
 - a. In Name, specify Web Search Agent as the name of the agent.
 - b. In Role, specify Web Search Specialist as the role of the agent.
 - c. In Backstory, provide the following backstory You are a reliable web search specialist who has a knack for finding great information online using a web search tool. You are an expert in creating comprehensive search phrases that yield the best search results.
 - d. In Goal, provide the following goal Return a summarized set of bullets that cover the most relevant details for the search required.
 - e. Click Create or Edit Tools. The Create or Edit Tools page displays.
 1. Click Add Tool.
 2. Click Add from Template.
 3. Click Close.
 2. Click Create Agent to create the agent.
 3. Click Close after saving the agent.
- c. Click Create Agents and create the second Agent with the following details:
 1. In Name, specify Creative Monetization Strategist as the name of the agent.
 2. In Role, specify Creative Monetization Strategist as the role of the agent.
 3. In Backstory, provide the following backstory You are a creative agent who reviews the information provided as an input and then comes up with ways to make money using this information.
 4. In Goal, provide the following goal Provide a list of 5 bullets in markdown using the information you are given. This agent does not need any tools.
 5. Click Create Agent to create the agent.
 6. Click Close after saving the agent.
- d. Once both agents are created, click Save & Next in the Add Agents page to proceed with the next steps.

- e. In the Tasks page, follow the steps outlined below. Since this is not a conversational workflow, you will need to define specific tasks for each agent to accomplish. Ensure that tasks are created in the correct sequence to maintain a logical workflow.



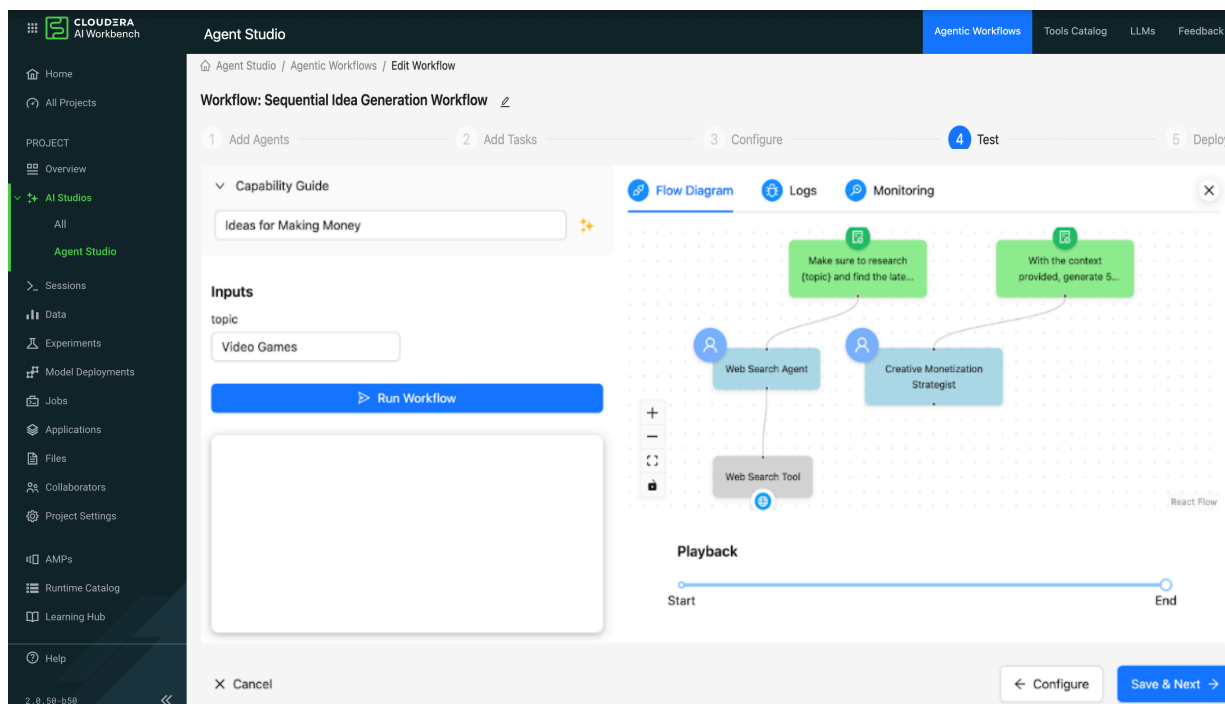
1. Set the following parameters for the first task:
 - a. In Task Description, provide the description as Make sure to research {topic} and find the latest information in the field.
 - b. In Expected Output, provide the expected out put as Provide a list of outputs with interesting facts from what you find about {topic}. Keep your answer in the markdown format.
 - c. In Select Agent, select Web Search Agent.
 - d. Click Add Task to include this task in the workflow.
2. Set the following parameters for the second task:
 - a. In Task Description, provide the description as With the context provided, generate 5 ideas on ways to make money using that information.
 - b. In Expected Output, provide the expected out put as 5 bullets formatted in Markdown
 - c. In Select Agent, select Creative Monetization Strategist.
 - d. Click Add Task to include this task in the workflow.
- f. Click Save & Next after adding both tasks to proceed to the next step in the workflow configuration.

9. In Configure page, set the following parameters:



- a. In Max New Tokens, specify 1024. This value represents the maximum number of new tokens the agents and manager agent can generate during LLM calls. Note that certain LLM endpoints may impose restrictions on this value.
- b. Set the Temperature to 0.2. The temperature controls the variability or creativity in the agent's responses generated by the LLM. A higher temperature value results in more diverse and creative responses. A lower temperature value results in less varied and more deterministic responses.
- c. Under Tools, in serper_api_key, specify your Serper API key.
- d. Click Save & Next.

10. In the **Test** page, you can choose a topic of your choice to test the workflow. As an example, Video Games is selected.



- a. Click **Run Workflow** to execute a test of the configured workflow.
- b. Once the test is complete, click **Save & Next** to proceed to the deployment stage.

11. In **Deploy** page, you can perform the following :

- Click **Deploy** to initiate the deployment of the workflow. The deployment process typically takes 5 to 7 minutes to complete. Once deployed, navigate to the **Applications** tab in the left navigation pane to view the deployed workflow.
- Click **Save Template** to create a reusable template of the workflow. This template can serve as a reference for future use.
- Click **Test** to return to the previous page and re-test the workflow, if necessary.