



## **Agent Studio User Guide v2 (external)**

**July 2025**

## Table of Content

1. Cloudera Agentic AI Workflow: Story build up
2. High Level Architecture - Agent Studio Use-case 1: Insurance
3. Lab Setup and Steps - Let's Start!!
4. Let's run the workflow

## 1. Cloudera Agentic AI Workflow: Story build up

I recently stepped into a new chapter as an **Insurance Advisor** – not just with one, but two major players in the industry: **Star Health Insurance** and **ICICI Lombard**.

At our last team sync, my manager dropped a mission-critical task on my plate:

"Reach out to all your customers and **recommend insurance upgrades tailored to their profiles.**"

Now let me tell you – this wasn't just a casual outreach job.

I'm talking to hundreds **of customers**. Each one with their own policies, risk levels, family situations, cities, payment methods – you name it.

I was like, "Cool... so you want me to manually read through hundreds of profiles and come up with personalized insurance advice?!"

That's when it hit me: this is exactly the kind of chaos AI was made for.

So I rolled up my sleeves and got to work and stumbled onto something game-changing:

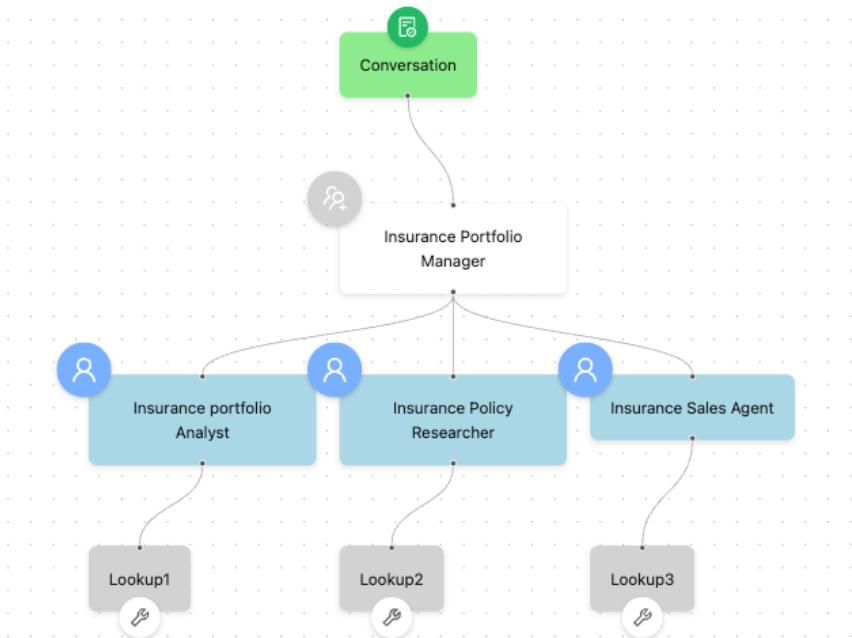
An **Agentic AI Workflow**, built on **ClouderaAI**

## 2. High Level Architecture – Agent Studio Use-case 1: Insurance

**An Agentic AI Workflow, built on ClouderaAI.**

With a bit of tooling and some logic, built an intelligent pipeline that could:

-  **Pull real-time customer data** securely from Hive on Cloudera
-  **Analyze policy holdings** across both **Star Health** and **ICICI**
-  **Match customers to relevant upgrades** based on their unique profile and risk factors
-  **Deliver personalized recommendations**—automatically, at scale
-  **Protect sensitive data** by design (no access to credit card details, ever)

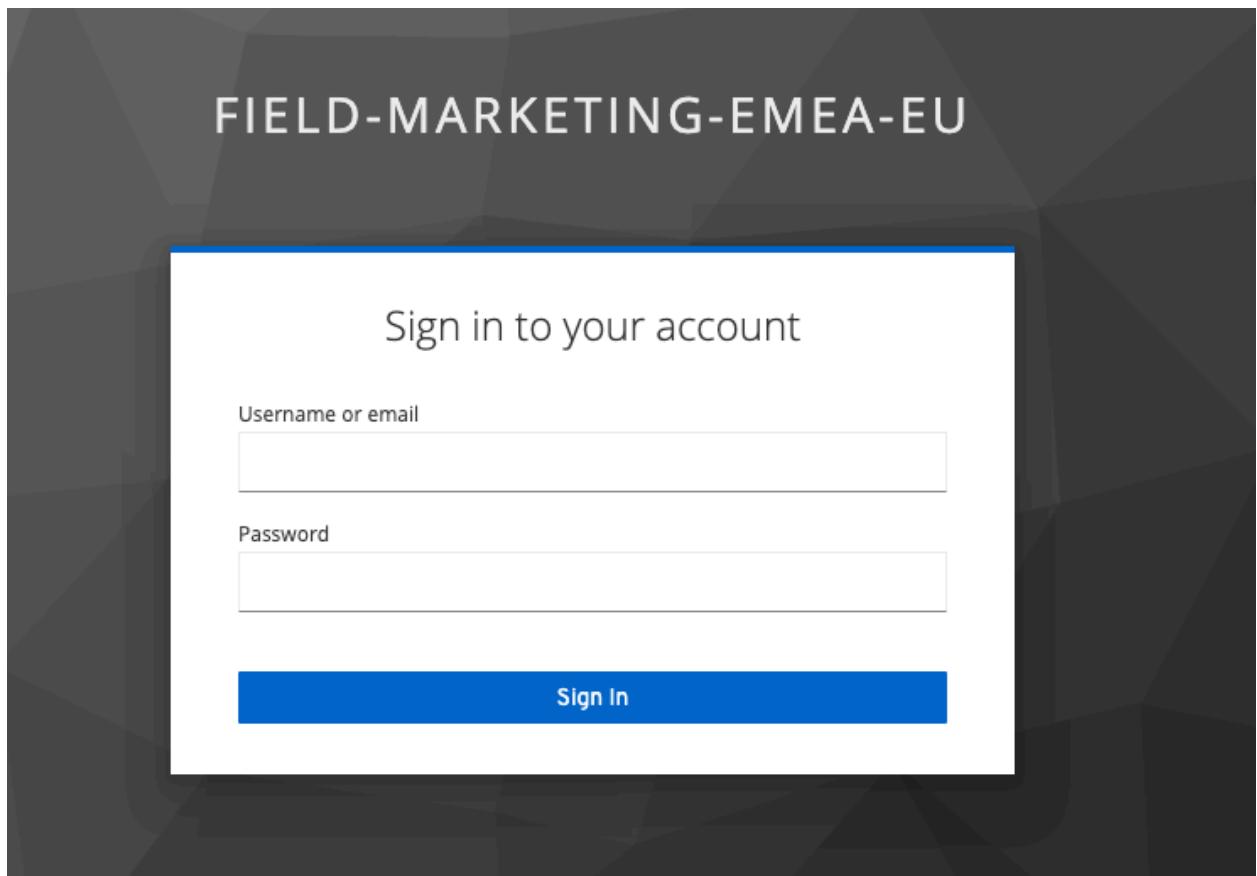


### 3. Lab Setup and Steps – Let's Start!!

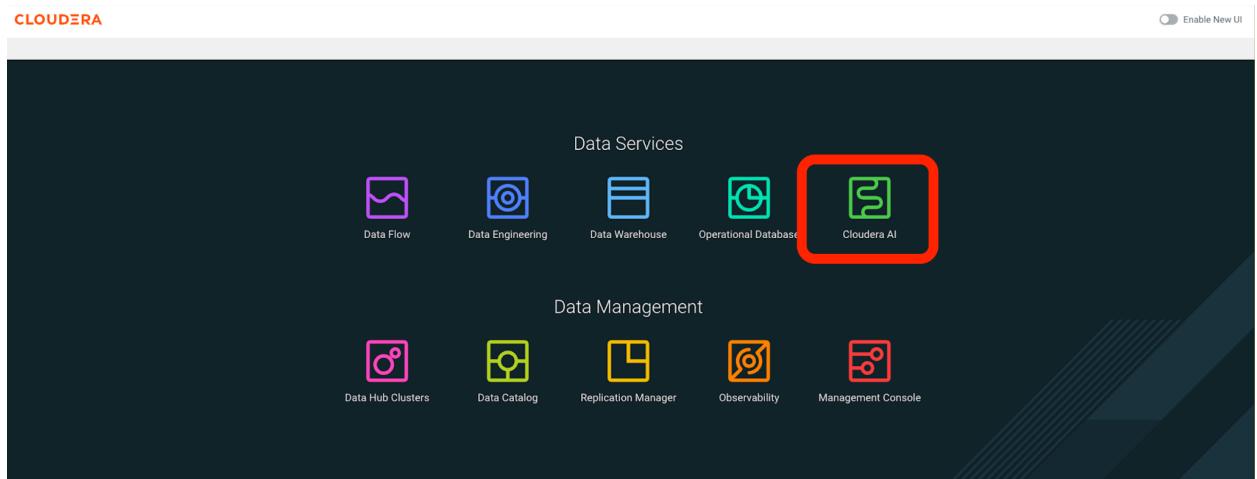
**Note:** Environment and Credentials will be shared by the Presenter.

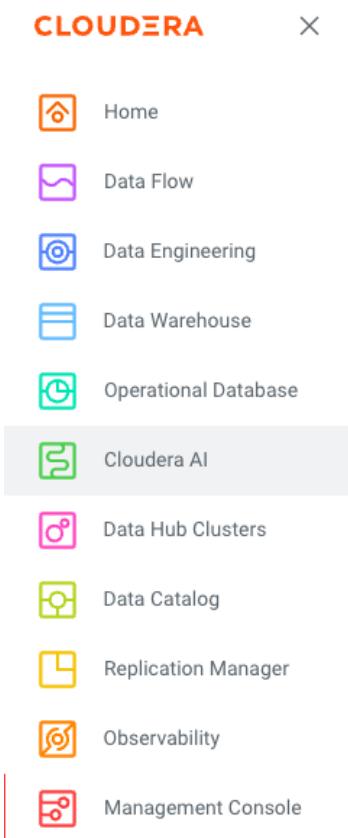
#### Step 1: Log in to Cloudera Platform & browse through Agent Studio UI

1. Log in using the credentials provided. **Please use an Incognito window.**



## 2. Select Cloudera AI





3. Click on **Workbench** > Select agentic-ai-hol-userxx-xx (example: user01-01 workbench).

**Note:** Please use this table to select the right workbench

User	Workbench
User001-010	agenic-ai-hol-user01-10
User011-020	agenic-ai-hol-user11-20
User021-030	agenic-ai-hol-user21-30
User031-040	agenic-ai-hol-user31-40

The screenshot shows the Cloudera AI Workbenches interface. On the left is a sidebar with the Cloudera AI logo at the top. Below it are three main sections: AI HUB, DEPLOYMENTS, and ADMINISTRATION. Under ADMINISTRATION, the 'AI Workbenches' option is selected, indicated by a green vertical bar on the left and bold text. The main content area is titled 'Cloudera AI Workbenches' and contains a search bar with placeholder 'Search Workbenches', a 'Environment' dropdown set to 'All', and a table of workbenches.

Status	Version	Workbench
Suspended	2.0.50	agentic-ai-hol-user01-10
Suspended	2.0.50	agentic-ai-hol-user21-30
Ready	2.0.50	agentic-ai-hol-user11-20
Suspended	2.0.50	Cloudera-AI-Workshop

4. Go to “**Projects**” on the left side panel > Select “**Public Projects**” from the dropdown.

The screenshot shows the Cloudera AI Workbench Home page. On the left sidebar, 'Projects' is selected. The main area features a 'Create a new project' section with a 'Create' button and a 'Deploy a prototype' section with a 'Deploy' button. Below these are 'Recent Projects' and 'Product Tour' sections. The 'Recent Projects' section displays a placeholder message: 'You currently don't have any projects. Select an option above to get started.' At the bottom, there's a 'View all projects >' link.

The screenshot shows the Cloudera AI Workbench Projects page. The left sidebar has 'Projects' selected. The main area includes a 'Search Projects' bar, a 'Scope' dropdown set to 'Public Projects', and a 'Creator' dropdown set to 'All'. A modal window titled 'Public Projects' is open, listing two projects: 'user011' and 'user013', both owned by 'asoni'. The 'Public Projects' tab in the modal is highlighted.

5. Click on the publicly available project named **HOL\_Agent\_Studio**

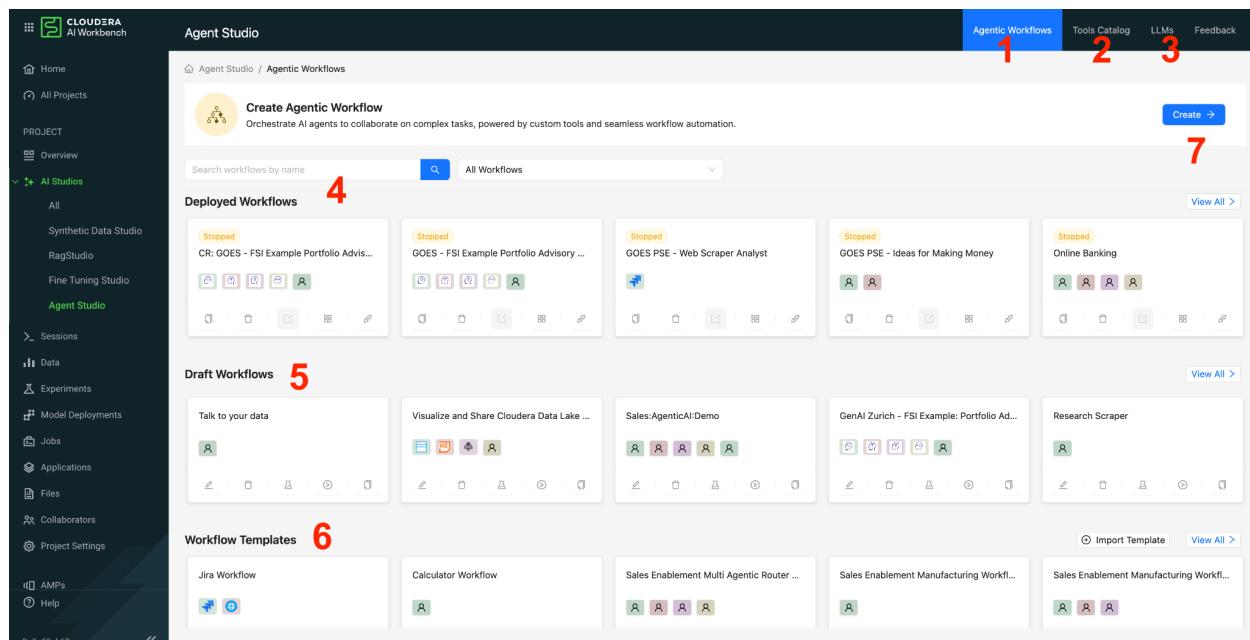
The screenshot shows the Cloudera AI Workbench interface. On the left, a sidebar menu includes Home, ALL, Projects (which is selected), Sessions, Experiments, Model Deployments, AI Registry, Jobs, Applications, AMPs, Runtime Catalog, Learning Hub, User Settings, Help, and a version number 2.0.47-b365. The main content area is titled "Projects" and shows a search bar, a scope dropdown set to "Public Projects", and a sort dropdown set to "Last Updated". A table lists one project: "HOL\_Agent\_Studio". The project details show it was owned by "sbalachandar" and last worked on "an hour ago". At the bottom right of the table, there are navigation buttons for page 1 of 25.

6. Click on **AI Studios > Agent Studio**

The screenshot shows the Cloudera AI Workbench interface. The sidebar on the left has sections for PROJECT (Overview, AI Studios, All, Synthetic Data Studio, RagStudio, Fine Tuning Studio, Agent Studio), SESSIONS, DATA, EXPERIMENTS, MODEL DEPLOYMENTS, JOBS, and APPLICATIONS. The "Agent Studio" item under PROJECT is highlighted with a red dashed box. The main content area shows a header "shameed / HOL\_Agent\_Studio" and a sub-header "HOL\_Agent\_Studio". Below this is a "Models" section with a table:

Model Deployment	Source	Status	Replicas
GOES - FSI Example Por_TxpN1ckQ	[src/eng...]	Deployed	1 / 1
Mert - Civilian Las Ve_40GyyBwJ	[src/eng...]	Deployed	1 / 1
Mert - FSI Example: Po_bBWFtH2O	[src/eng...]	Deployed	1 / 1
AirAware - Analyze the_GMzwmwSaL	[src/eng...]	Stopped	0 / 1
GOES PSE - Web Scraper_NRYkuxDU	[src/eng...]	Stopped	0 / 1
GOES PSE - Ideas for M_w2hVTvRR	[src/eng...]	Stopped	0 / 1
Online Banking_ka3L6seJ	[src/eng...]	Stopped	0 / 1
CR: GOES - FSI Example_aElxCDP	[src/eng...]	Stopped	0 / 1
airquality-time-series-adapter-llama-v1	[fine-tun...]	Stopped	0 / 1
Airquality Forecaster	[fine-tun...]	Stopped	0 / 1
fraud-detector-onnx-xgboost	[predict...]	Stopped	0 / 1

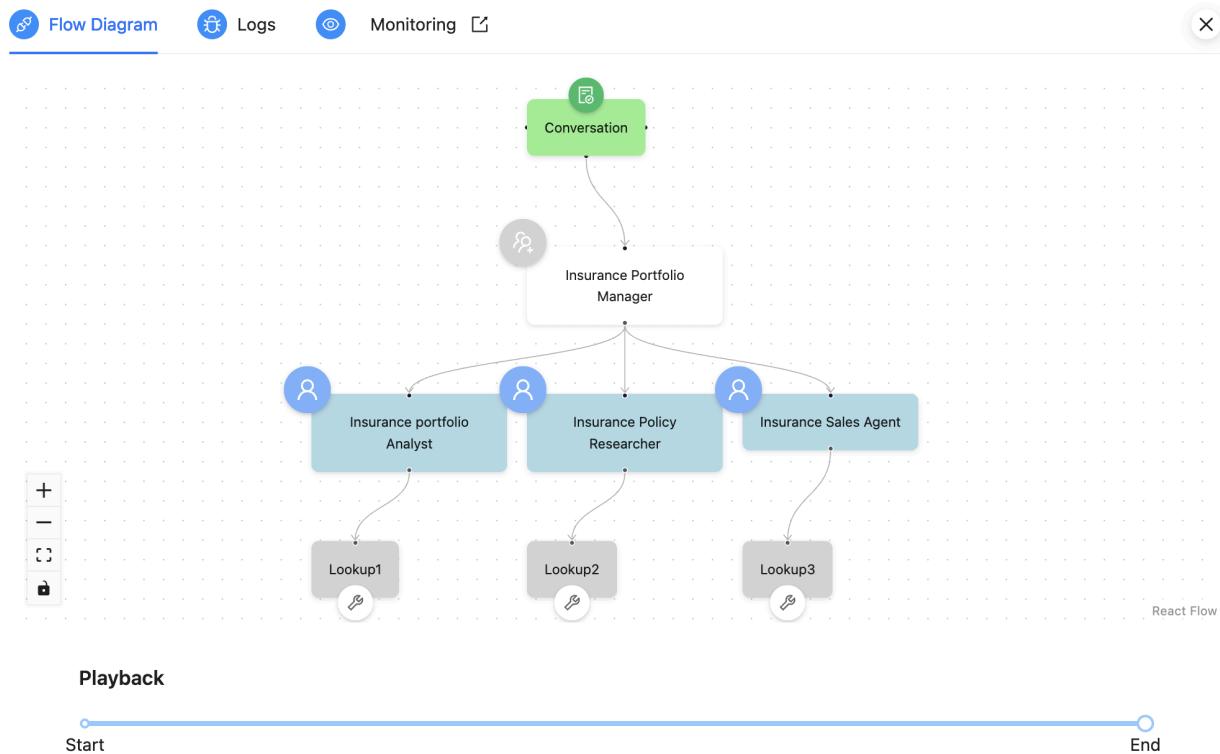
## Step 2: Understand the Agent Studio Environment



- 1. Agentic Workflows:** Dashboard with End-to-End Agentic Workflows & Use Cases Development and Deployment.
- 2. Tools Catalog:** Build custom Python tools to enhance your AI agents capabilities and supercharge your workflows.
- 3. LLMs:** Hybrid Muti LLM Model Registry to Register language models which will be used to build agents and workflows.
- 4. Deployed Workflows:** Single Pane of Glass to All Production Workflow and Use Cases.
- 5. Draft Workflows:** Management and Collaboration all the Development Workflows.
- 6. Workflow Templates:** Workflows Catalog to share and re-use the workflows.
- 7. Create:** Build your Agentic Workflow from scratch

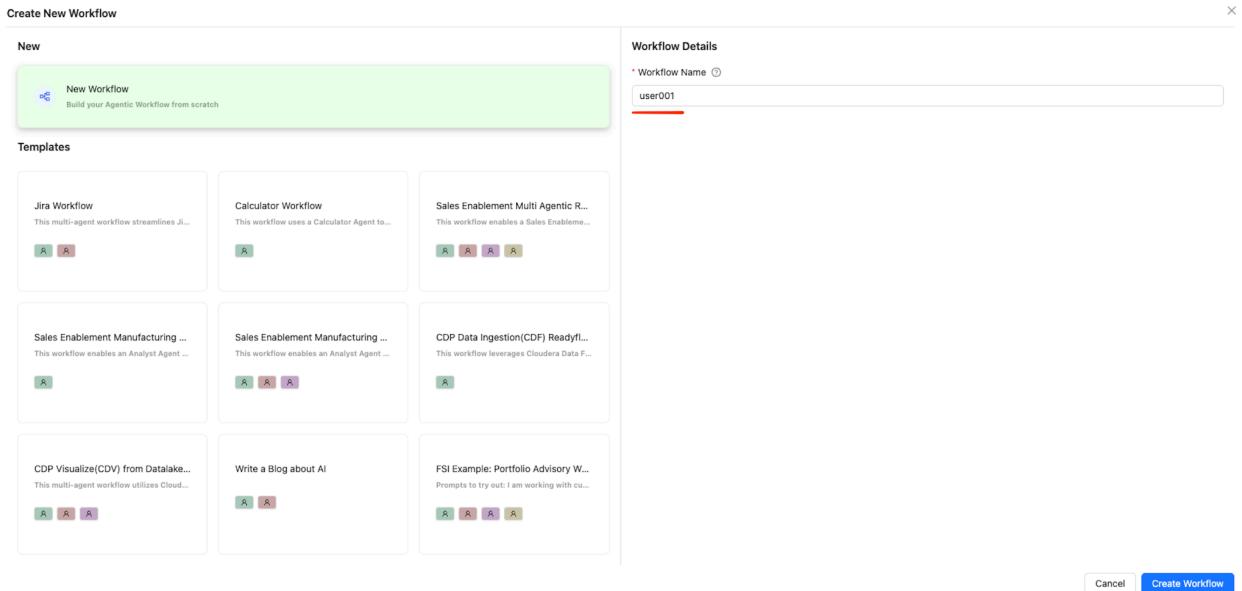
## Step 3: Let's create the workflow

Here is how your workflow would look like at the end



### 1. Create > Insurance Workflow:

- Select Create to get started
- Provide the workflow name as your username for example 'user001' and create agent



**c. Is Conversational: ON**

**d. Manager Agent: ON**

**e. Default Manager (click on configure custom manager):**

- **Name:** Insurance Portfolio Manager
- **Role:** Insurance Portfolio Manager
- **Backstory:** You are a Sr. manager capable of delegating tasks related to portfolio management and optimisation. One task is done by one agent only. Do not send one request to more than one agent.
- **Goal:** Delegate the user request properly and forward relevant execution information to the user.  
One task is done by one agent only. Do not send one request to multiple agents.  
Send customer-related lookups to Insurance portfolio Analyst  
Send research or find external policies to Insurance Policy Researcher  
Send recommend and do upselling to Insurance Sales Agent
- **LLM model:** Default

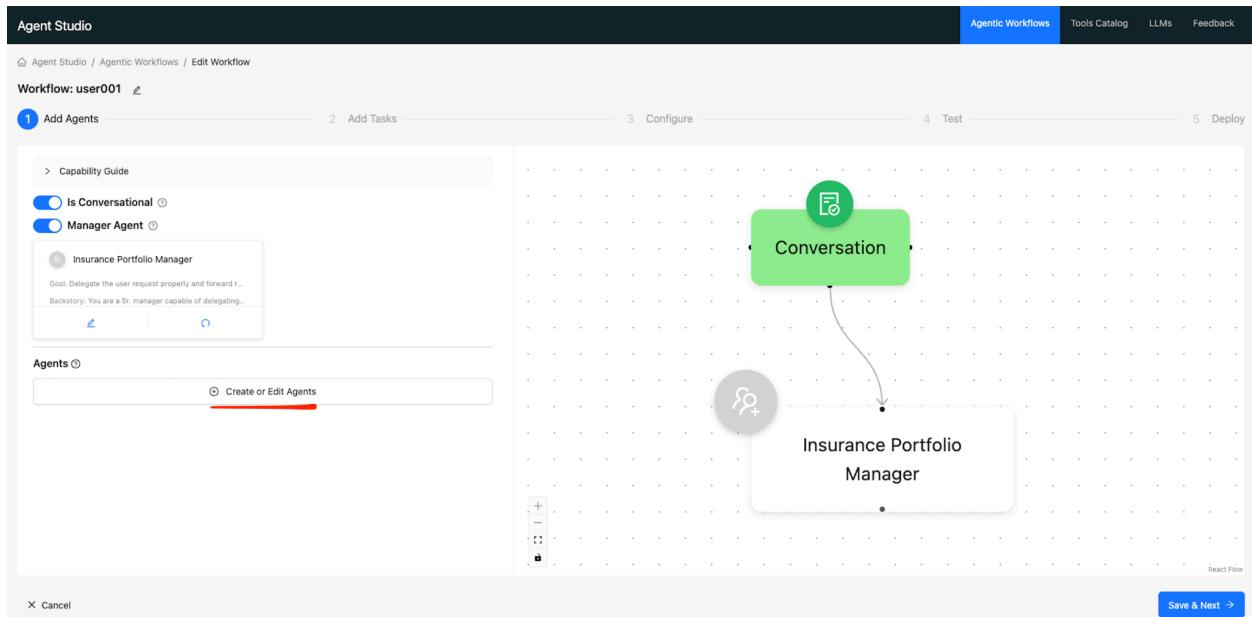
The screenshot shows the 'Manager Agent Details' configuration screen. At the top right are two buttons: 'Generate with AI' (blue) and 'Reset Fields' (light blue). Below these are several input fields:

- Name**: Insurance Portfolio Manager
- Role**: Insurance Portfolio Manager
- Backstory**: A text area containing: "You are a Sr. manager capable of delegating tasks related to portfolio management and optimisation. One task is done by one agent only. Do not send one request to more than one agent."
- Goal**: A list of actions:
  - Delegate the user request properly and forward relevant execution information to the user.
  - One task is done by one agent only. Do not send one request to multiple agents.
  - Send customer-related lookups to Insurance portfolio Analyst
  - Send research or find external policies to Insurance Policy Researcher
  - Send recommend and do upselling to Insurance Sales Agent
- LLM Model**: Azure Open AI (Default)

At the bottom right are 'Cancel' and 'Add Manager Agent' buttons.

**Note:** There is an option of Generate with AI. We won't be using this option today, but if you are creating another Manager Agent this could help!

## 2. Create Three Agents: (click on create or edit agents)



### a. Agent 1 : Insurance portfolio Analyst

- Click on Create or Edit Agent (as per above screen capture)
- **Name:** Insurance portfolio agent
- **Role:** Insurance portfolio Analyst
- **Backstory:** You are an experienced portfolio analyst who excels at understanding customer needs and current holdings.  
You are able to fetch their current holdings and investment profile. do not hallucinate and provide the details retrieved from hive DB
- **Goal:** Fetch any requested customer portfolio or profile data. Do not hallucinate and provide the details fetched from hive DB
- **LLM model:** Default

## Agent Details

Generate with AI

Reset Fields

\* Name

Insurance portfolio agent

\* Role

Insurance portfolio agent

\* Backstory

You are an experienced portfolio analyst who excels at understanding customer needs and current holdings.

You are able to fetch their current holdings and investment profile. do not hallucinate and provide the details retrieved from hive DB

\* Goal

Fetch any requested customer portfolio or profile data. Do not hallucinate and provide the details fetched from hive DB

\* LLM Model

Azure Open AI (Default)

## Add Optional Tools

+ Create or Edit Tools



No data

## Add Optional MCP Servers

+ Add MCP Server to Agent



No data

Close

Create Agent

- Click on Create or Edit Tools (as per above screen capture)
- Search for Hive Tool under Create Tool From Template

The screenshot shows the 'Create or Edit Tools' interface. On the left, there's a 'Create New Tool' button and an 'Edit Agent Tools' section with a search bar and a placeholder 'No data'. On the right, there's a 'Create Tool From Template' section with a search bar containing 'Hive Tool'. A red box highlights the 'Hive Tool' entry in the results list, which includes a tool icon, the name 'Hive Tool', a green checkmark, and the status 'N/A'.

- Select Hive Tool and click Create Tool from Template
- Click on Save Tool followed by clicking on Close and Save Agent and Close

The screenshot shows the 'Create or Edit Tools' interface with the 'Tool Details' tab open. It shows the tool name 'Hive Tool', a tool icon, and a playground section. Below that is a code editor window titled 'tool.py' containing the following Python code:

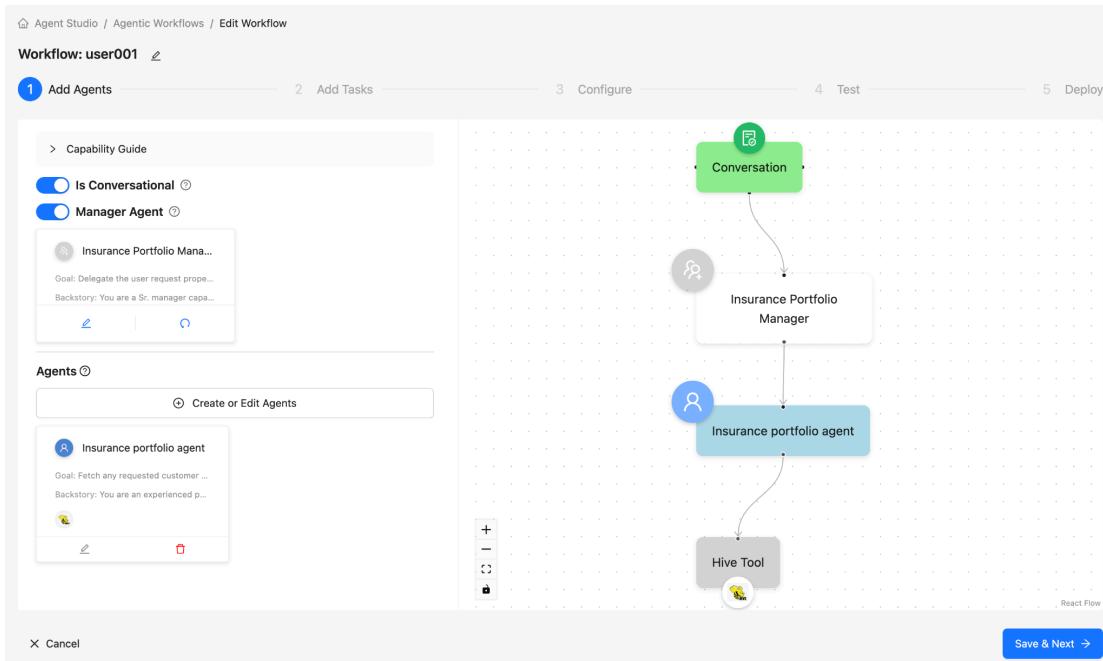
```

1 import os
2 from pydantic import BaseModel, Field
3 from pydantic import BaseModel as StudioBaseModel
4 from typing import Literal, Type, Optional
5 import json
6
7 CONNECTION_NAME = "default-hive-aws"
8
9 class UserParameters(BaseModel):
10     pass
11
12
13 class CustomerLookupTool(StudioBaseTool):
14     class ToolParameters(BaseModel):
15         customer_id: int = Field(description="customer ID to do lookup")
16
17     name: str = "Customer Lookup Tool"
18     description: str = "A simple tool to look up customer info"
19     args_schema: Type[BaseModel] = ToolParameters
20     user_parameters: UserParameters
21
22     def _run():

```

At the bottom right of the code editor, there are 'Edit' and 'Refresh' buttons, and below them are 'Close' and 'Save Tool' buttons. The 'Save Tool' button is highlighted with a red box.

- Here is how your canvas should look and let's proceed to add the next agent.



### b. Agent 2: Insurance Policy Researcher

- Click on Create or Edit Agent (as we did while creating agent 1)
- **Name:** Insurance Policy Researcher
- **Role:** Insurance Policy Researcher
- **Backstory:** This agent is designed to perform research on all available insurance policies from Star Health via `serper_api_key`. It should access brochures or policy documentation from Star Health's website and use it to identify additional policies, benefits, and coverage options that might be suitable for the customer.

- **Goal:** Responsible to analyse available policy documents and brochures.
- **LLM model:** Default
- Select Create or Edit Tools

Agent Details
 Generate with AI
 Reset Fields

---

\* Name ?

\* Role ?

\* Backstory ?

This agent is designed to perform research on all available insurance policies from Star Health via serper\_api\_key. It should access brochures or policy documentation from Star Health's website and use it to identify additional policies, benefits, and coverage options that might be suitable for the customer.

\* Goal ?

\* LLM Model ?

Azure Open AI (Default)

Add Optional Tools

+ Create or Edit Tools

K E - I - A - S -

Add Optional MCP Servers ?

+ Add MCP Server to Agent

K E - I - A - S -

Close

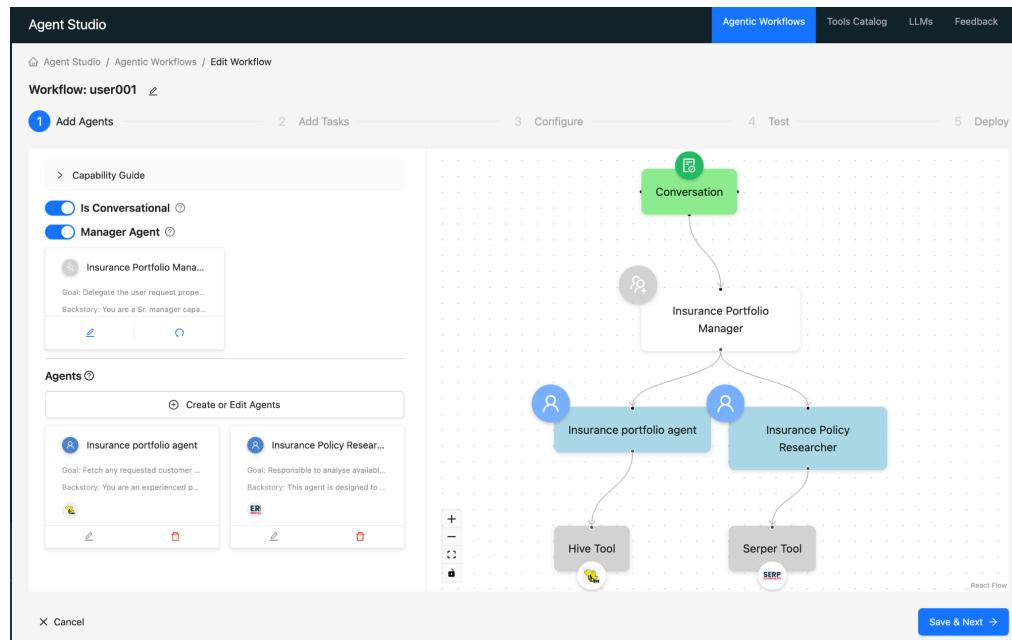
Create Agent

- Search for Serper Tool under Create Tool From Template

## Create or Edit Tools

The screenshot shows the 'Edit Agent Tools' section with a search bar and a message 'No data'. A modal window titled 'Create Tool From Template' is open, containing a search bar with 'Serper Tool' typed in, a result card for 'Serper Tool' with a green checkmark, and a note 'N/A'.

- Select Serper Tool and click Create Tool from Template
- Click on Save Tool followed by clicking on Close and Save Agent and Close
- Here is how your canvas should look and let's proceed to add the next agent.



### c. Agent 3: Insurance Sales Agent

- Click on Create or Edit Agent (as we did while creating agent 1)
- **Name:** Insurance Sales Agent
- **Role:** Insurance Sales Agent
- **Backstory:** To recommend personalised policy options to the customer. It should be able to propose new policies or suggest upgrades based on the customer's existing policies, risk profile, and coverage needs.
- **Goal:** Recommend a better insurance policy for the customer based on the current Star Health policy
- **LLM Model:** default
- Select Create or Edit Tools

Agent Details

Generate with AI Reset Fields

\* Name ?  
Insurance Sales Agent

\* Role ?  
Insurance Sales Agent

\* Backstory ?  
To recommend personalised policy options to the customer. It should be able to propose new policies or suggest upgrades based on the customer's existing policies, risk profile, and coverage needs.

\* Goal ?  
Recommend a better insurance policy for the customer based on the current Star Health policy

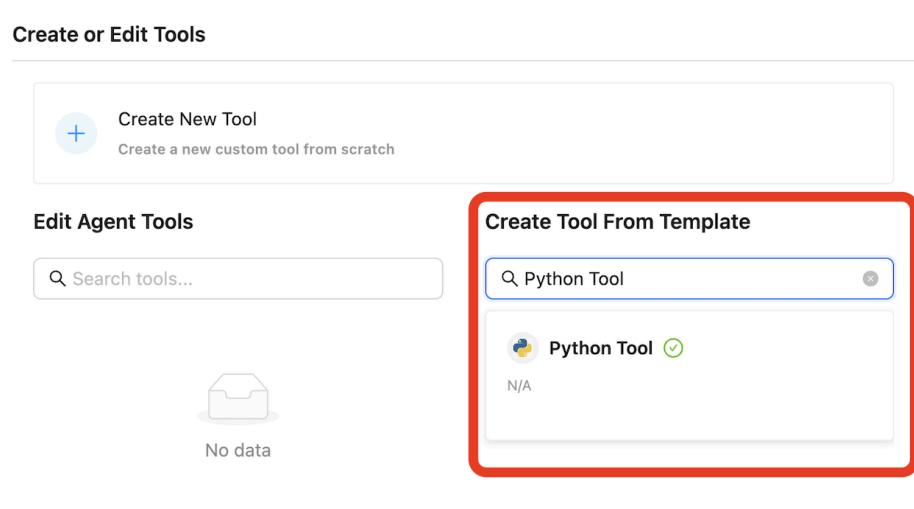
\* LLM Model ?  
Azure Open AI (Default)

**Add Optional Tools** + Create or Edit Tools

**Add Optional MCP Servers** + Add MCP Server to Agent

Close Create Agent

- Search for Python Tool under Create Tool From Template



- Select Python Tool and click Create Tool from Template
- Click on Save Tool followed by clicking on Close -> Save Agent -> Create Agent -> Close
- Here is how your canvas should look and let's proceed to add the next agent.

Workflow: user001

1 Add Agents    2 Add Tasks    3 Configure    4 Test    5 Deploy

**Agents**

- Insurance portfolio agent
- Insurance Policy Researcher
- Insurance Sales Agent

Save & Next →

## Step 4: Let's run the workflow

- Click on Save and Next

Workflow: user001

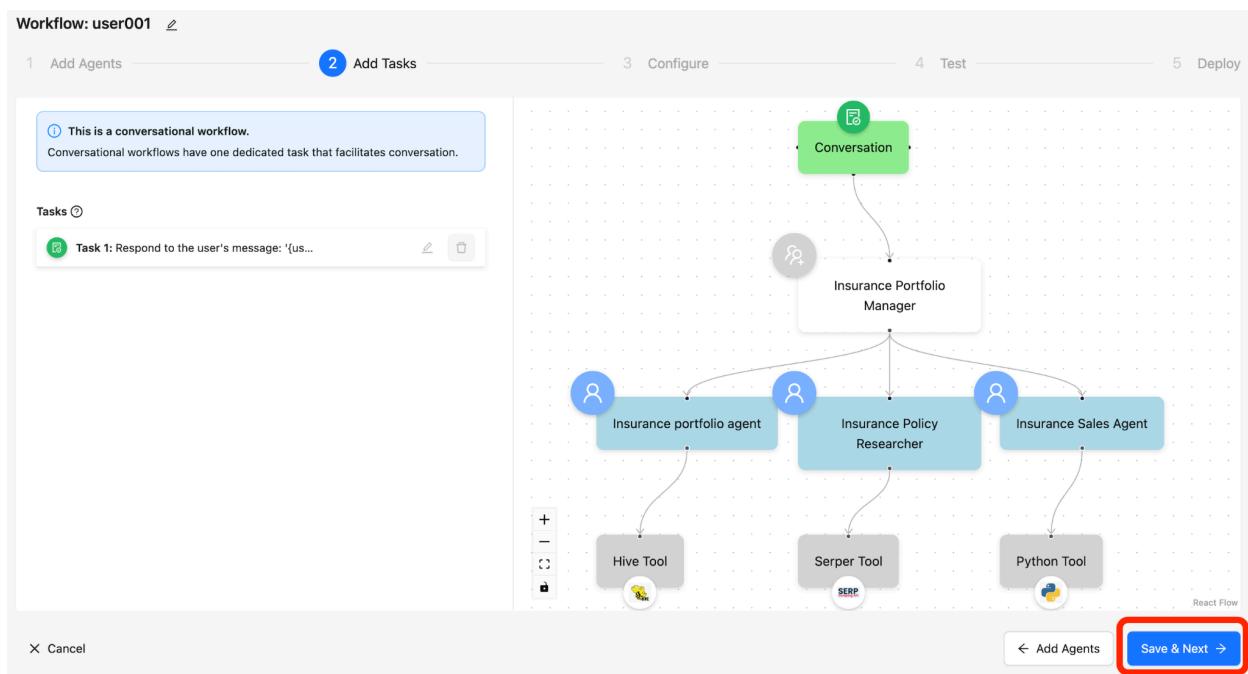
1 Add Agents    2 Add Tasks    3 Configure    4 Test    5 Deploy

**Agents**

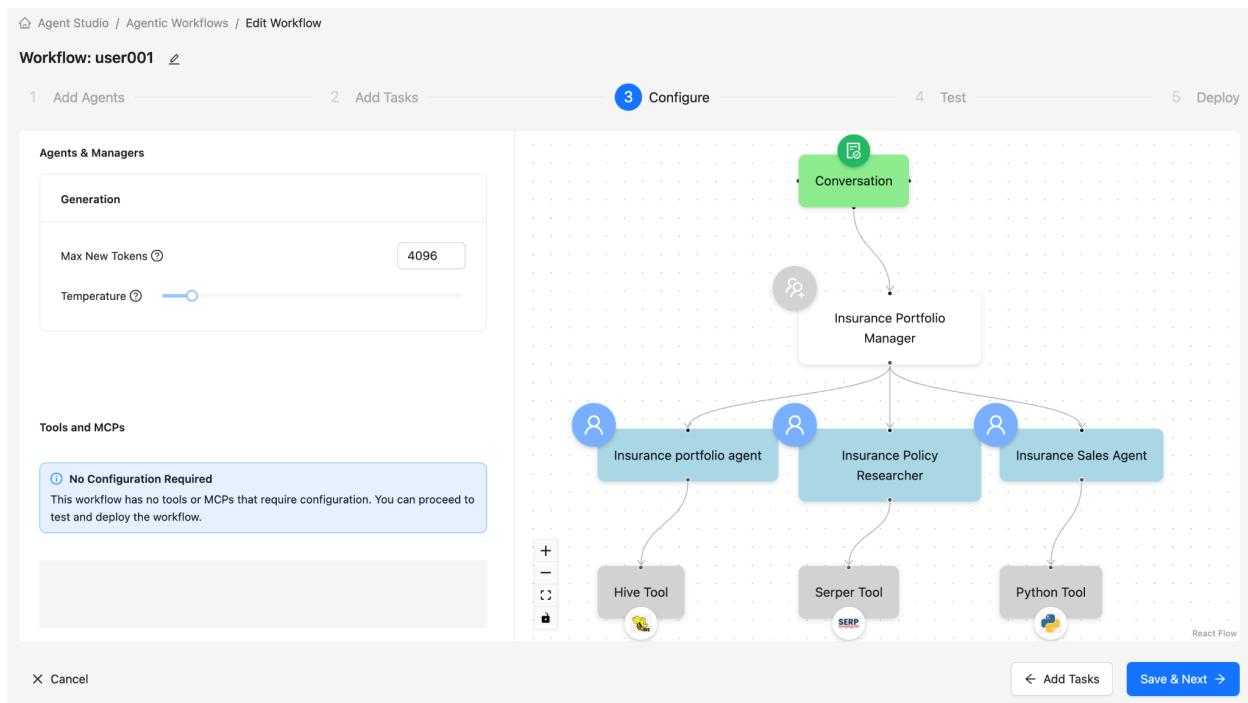
- Insurance portfolio agent
- Insurance Policy Researcher
- Insurance Sales Agent

Save & Next →

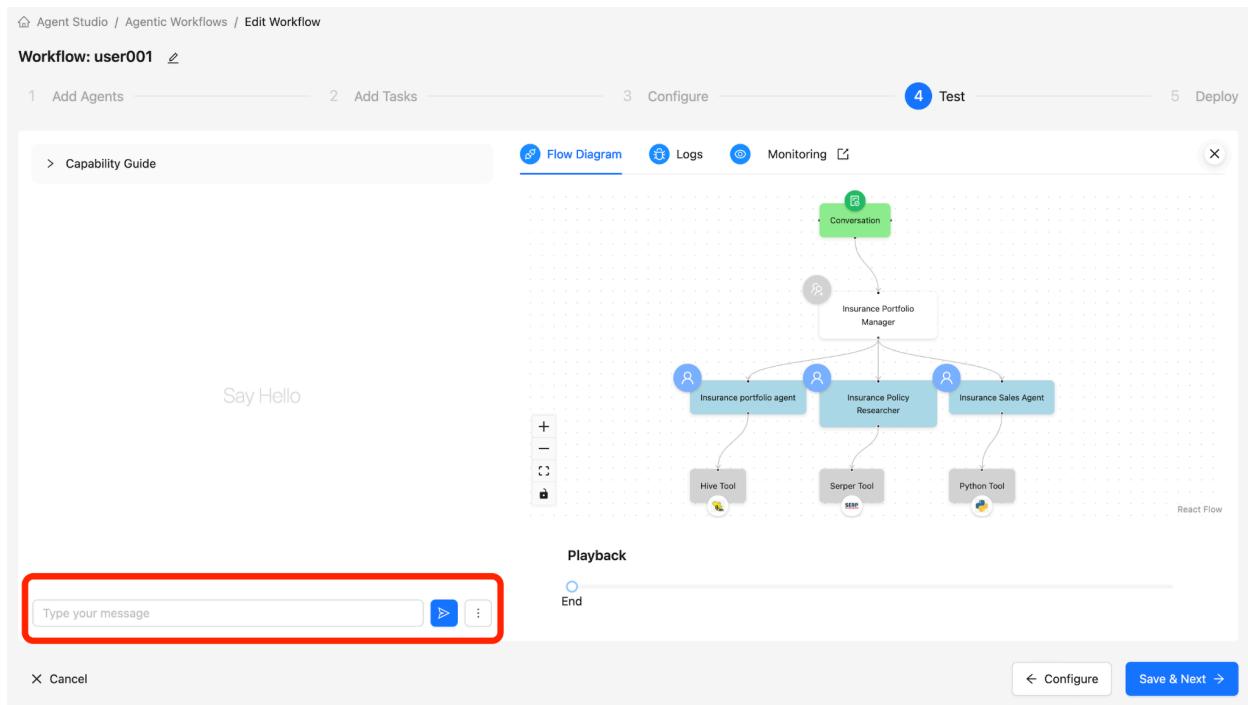
- Review and Click on Save and Next



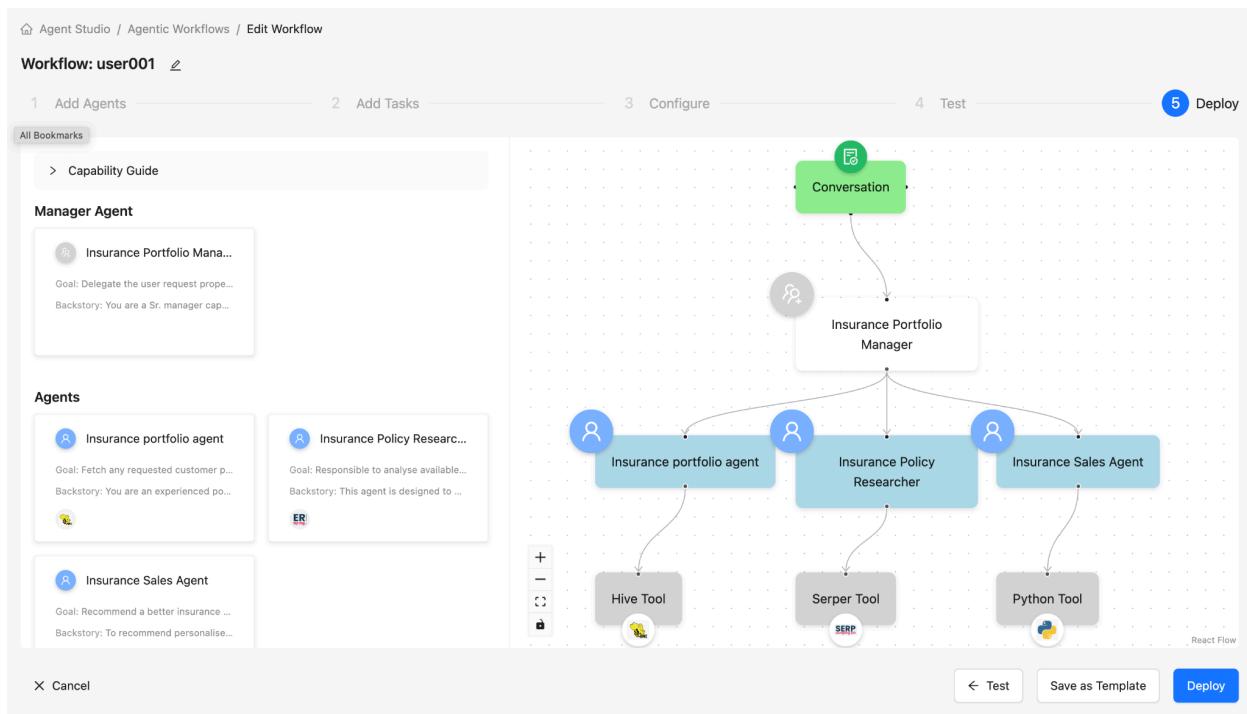
- Set the Temperature to 0.1 and Review and Click on Save and Next



- It's time to play with the agent



- TEST the workflow with the following queries:
  - **Prompt 1:** "Fetch customer details 600001" (Also could do 600001 -> 6000030)
  - **Prompt 2:** "Research other policy documents and brochures on star health"
  - **Prompt 3:** "Recommend a better insurance policy for <customer\_name> customer id <60000X> based on the current Star Health policy"
- You could deploy this into the Production workflow. But for now we will keep it simple and continue to run in the testing phase



- You are done with the Lab, you can continue testing or you can close the browser window

**CONGRATULATIONS!**  
YOUR USE CASE IS  
**READY**  
AND YOU HAVE  
**COMPLETED THE LABS!**

