Elements.cloud

# Configuring Agentforce Agents

Collaborate on the design of agents, build and test more quickly and provide governance

# Benefits of a process-led approach

**Before we jump in, here's a quick summary:** Elements.cloud is the platform of choice for implementing Agentforce, providing a clear and collaborative design approach.

- Quickly identify and validate agent use cases.
- Engage stakeholders to agree on scope.
- Provide confidence on how the agent works so that stakeholders can agree to deploy the agents.
- Design complete, working agents with clear logic and behavior.
- Identify and capture requirements for actions and use AI to generate user stories.
- Accelerate Agent development by automating instructions.
- Faster evaluation, testing, and debugging with AI-generated test cases.
- Provide a change log of diagram, instruction, and test case versions.
- Provide governance and versioning of agent designs.
- Control org complexity and tech debt with Agent metadata in the core metadata dictionary

## Prerequisites

Before implementing Agentforce with Elements.cloud:

- **Be a registered user with Elements**
  If you are not an Elements user, you can register for free. If you are a Salesforce employee, register and then click on your OKTA tile for access.
- **Salesforce Org Agentforce Licenses**
  Access to Agent Builder and Prompt Builder in your Salesforce Org
- **Elements.cloud Agentforce License**
  or Consulting License, or licensed through Salesforce: Access to build diagrams including AI-generated, build instructions and test utterances, and manage Agentforce metadata.
- **Edit rights in an Elements Workspace**
  If you are new to Elements, you can view this guide to understand and either join an existing Space or create a new one to work in. If you have been invited to a space, and cannot create content, you may not have edit rights on the space. You will need to click on the "Gear" icon top right of your screen,

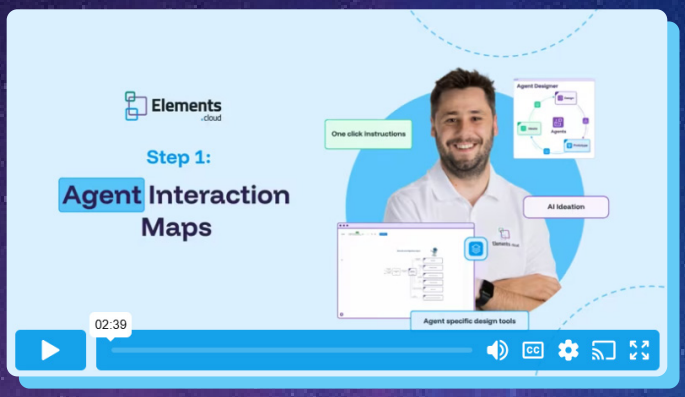

and select "Space settings". You will see the space admins listed who can allocate you with edit rights.

***Note 1:*** If you haven't seen this in action - this 4 min video shows the following approach applied to build and deploy an Agent that is Discovery Framework Coach. It has 5 actions and 26 steps to direct the Agent's reasoning and planning. In less than a week, it went from idea to production and is live in our business and working consistently.

***Note 2:*** If you are interested in using Variables and Filters to route and manage more sophisticated agents within Agentforce, please contact us to discuss your use-case. We are getting good results by incorporating them into this approach and there are nuances worth discussing on a web meeting. More guidance will follow in the coming weeks.

Step 1

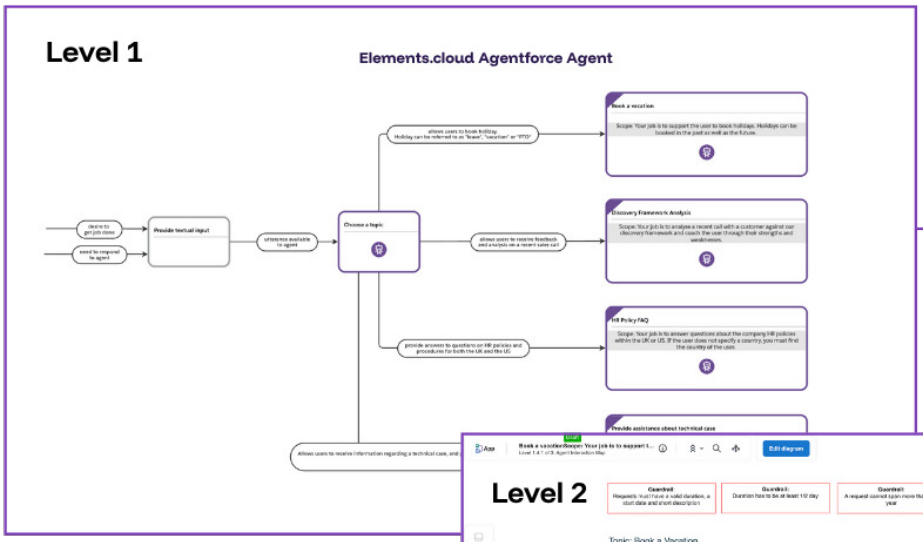
# Document Agent Use Cases and Jobs to be Done



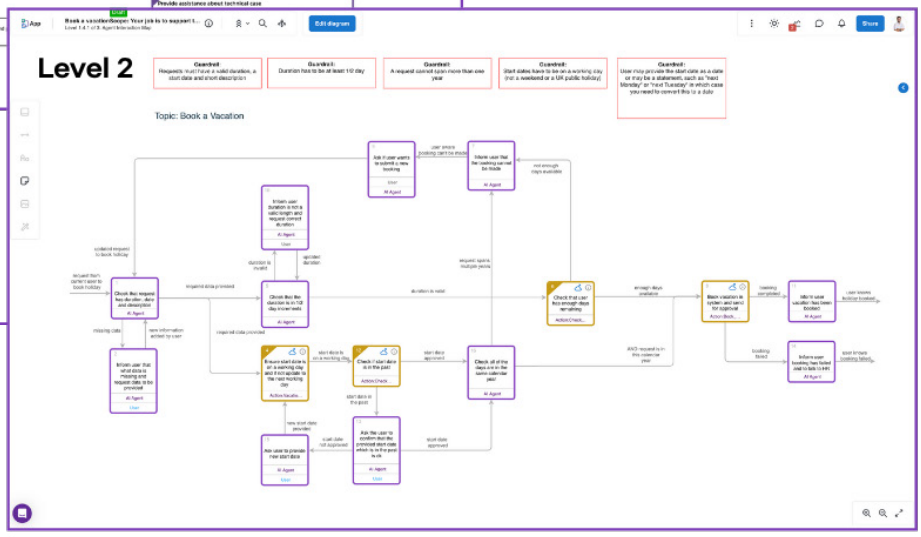

## 1.1 Understand “Agent Interaction Map” (AIM) structure

The first step is to create [Agent Interaction Map (AIM)] for the agent you are designing.

**The map has 2 levels:**




**Level 1**
List of Jobs To Be Done (JTBD) - which equate to Topics to Agent Builder



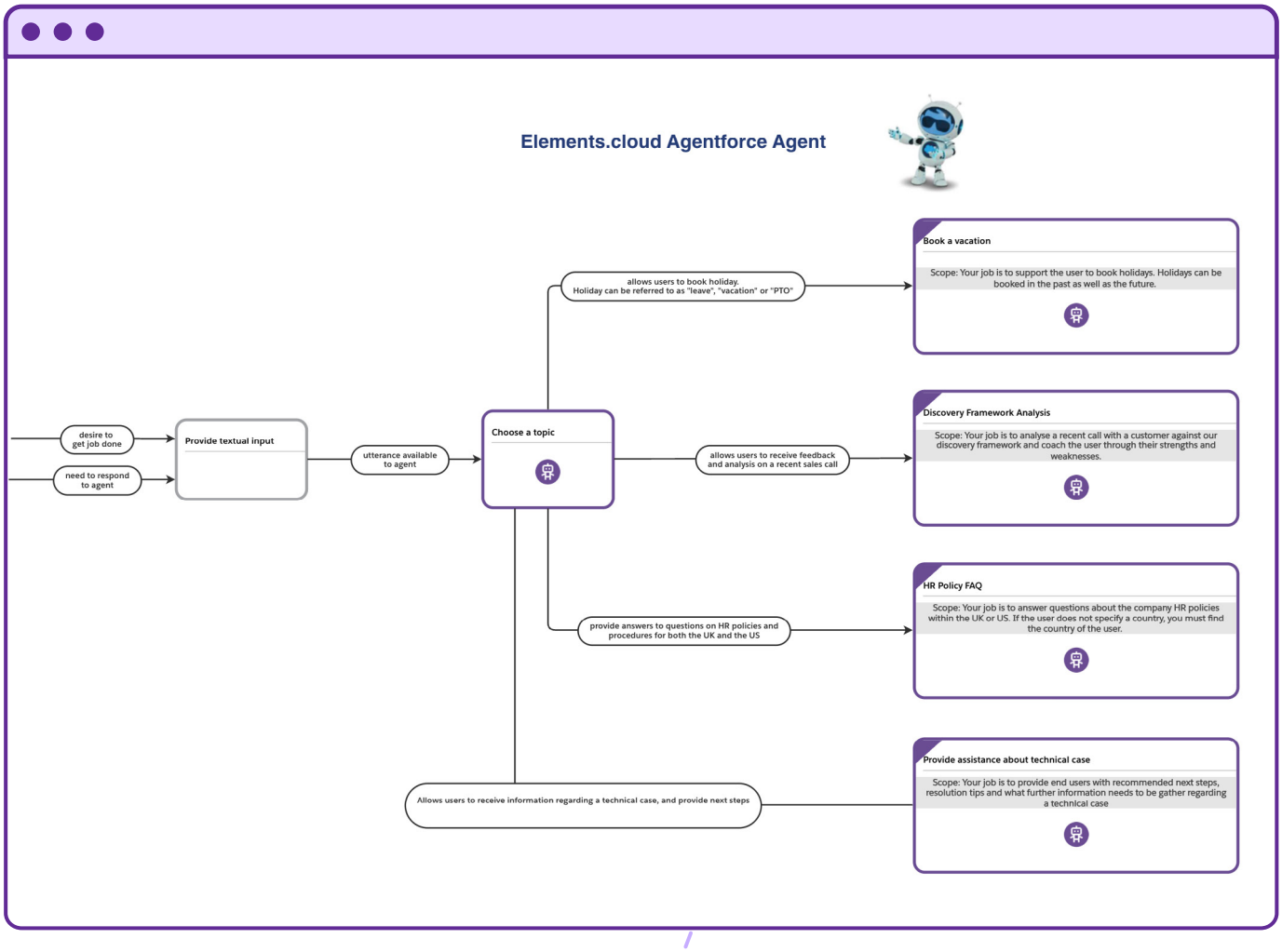

**Level 2**
Agent Instruction Diagram (AID) that is the JTBD. This describes the actions and the overall process flow with instructions.

# 1.2 Create AIM (Level 1) diagram

This top-level diagram shows the user interaction and on the right side of the diagram, the boxes represent the Topics. How the Atlas Reasoning Engine decides is based on the Topic Classification Description and Scope. You can see the Topic Classification description is the text on the line going into the JTBD / Topic box, and the Topic Scope is the text in the box.

You want to make the Topics tight and self-contained so that you don't confuse the Atlas Reasoning Engine. For example, you don't want all the instructions and actions for Booking, Amending, Cancelling Vacation, and Checking Availability in one Topic. These would be 4 separate Topics.

You can copy the blank AIM template from the Elements Public Templates workspace to use as a starting point. It has the structure and also the color scheme.




*Agent: Level 1 diagram of AIM*

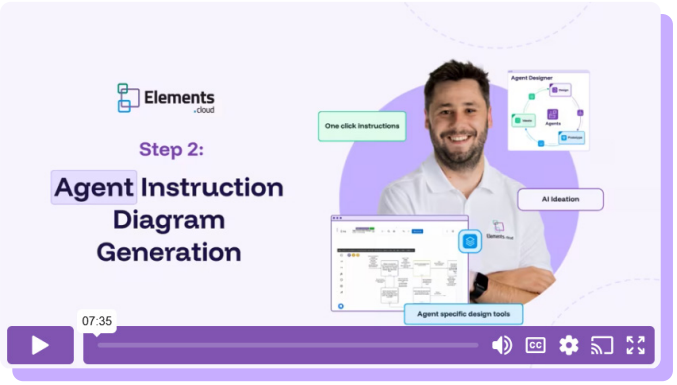
Step 2

# For Job to Be Done (JTBD), create the Agent Instruction Diagram (AID)

## 2.1 Creating the Agent Instruction Diagram

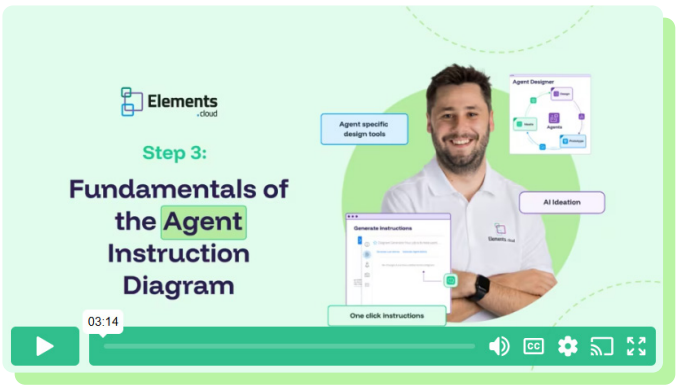
The next step is to create a Level 2 diagram, called an Agent Instruction Diagram (AID). It is a drill down from a box in the Level 1 diagram.

This is the Jobs To Be Done (JTBD) diagram that describes the Topic.




**There are 4 ways to create that JTBD Level 2 diagram.**

1. You can **copy a template** or existing diagram.
2. Elements AI will **draw a diagram from your org metadata.**
3. You could use Elements AI to **draw the diagram from a sketch**, image, diagram, script or SOP.
4. You can **start with a blank sheet** in a live workshop with your stakeholders.



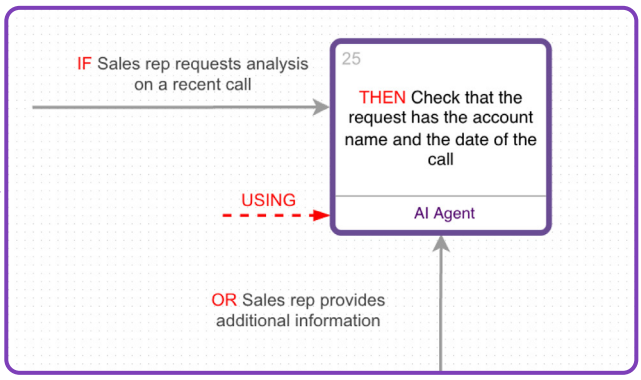

## 2.2 Refining your JTBD / AID

Whether you are using templates or starting from a blank screen, you should review this short video which explains the fundamentals of creating the most effective and reliable Agent Instruction Diagrams.

## 2.3 Key principles for drawing Agent Instruction Diagrams

- The lines are the input and the activities are the instructions. So they should read “IF {input text} THEN do {activity text}.

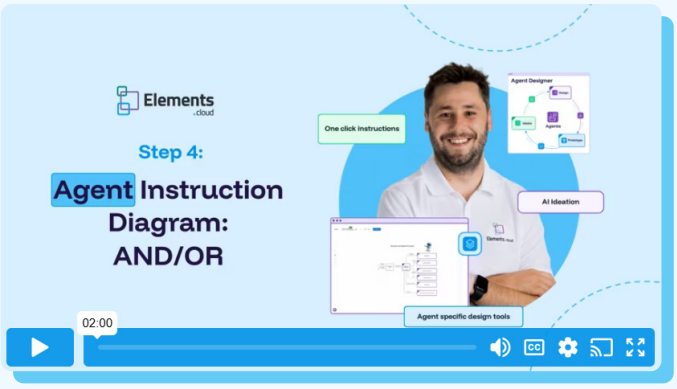
  ***NOTE:*** The red text is to show you what we mean, not to be written on the diagrams!




- The activity text should start with a verb that is specific. Avoid *Process* or *Manage*. If you are getting data, then *Get* or *Fetch* are better verbs than *Find*. Be unambiguous; e.g. instead of “add” do you mean “update” or “create”
- If you have 2 lines going into a box, and the text is not identical then the agent will assume it is an OR
- If you want the 2 lines into a box to be AND, then start the 2nd text with AND
- Avoid having 2 different boxes with the same input text as it confuses the agent.
- The instructions are created for the agent based on the activity box number. You can automatically renumber the activity boxes. (see renumbering article)
- The activity boxes are either performed by *Agent*, *User* or *Action*. We suggest you use our color scheme so it is easy to read the diagrams. The colors have been chosen for maximum contrast (508 compliance). You can copy and paste the themes per box type from any one of our templates. (see colors styles and formats article)
- The resources on the activity boxes should be *Agent*, *User* or *Action*.
- Create user stories for each Action activity box. The action is delivered by Salesforce metadata - Apex, Flow, Prompt Builder. These may need to be created.

## 2.4 Understanding how to refine logic within your diagram

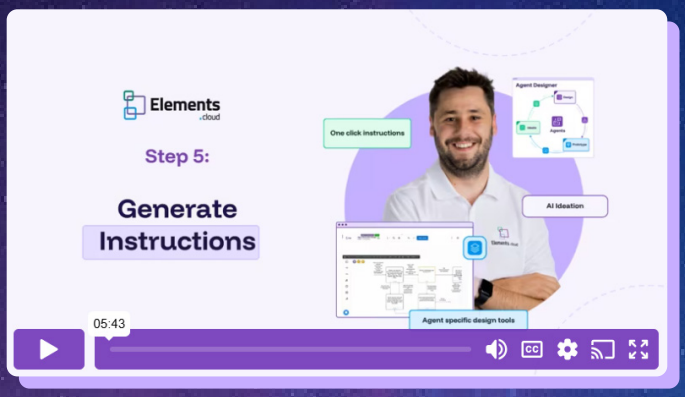
Capturing precise logic is important when deciding how you want to educate Agentforce with the deterministic paths. These are what you would like it to base its reasoning and planning.




Even though an agent is inherently probabilistic, you can still direct it, (in the same way you would with a human employee), by being very specific about when to follow certain paths and what to do. e.g. “IF” certain things happen or conditions are met. In this context, there is a simple but effective way to build AND/OR logic into these diagrams.

Step 3

# Create the agent




## 3.1 Create Agent in Salesforce Agent Builder

Your Level 2 AID will be a single Topic in Salesforce Agent Builder. This Topic could be added to a new or an existing Agent.

1. First, create an Agent, or use and existing Agent or the internal “Employee Agent” BTW It used to be called Einstein Copilot.

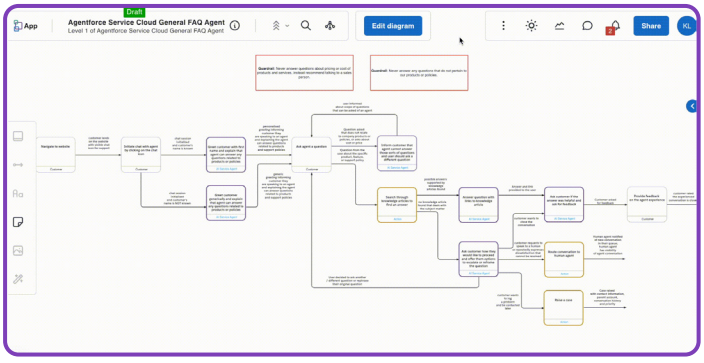
2. Then create a new Topic and give it a name. This should be same as your Level 2 AID. You can copy the “Classification” and “Scope” from the Level 1 diagram input and activity box text into the Topic in Agent Builder.

3. You need to create Actions for each Action activity box in your Level 2 AID. Each of these are Actions linked to Salesforce metadata - Apex, Flow, Prompt Template. If they do not exist, you need to create them.

All of this work is required before you add the Topic instructions.

## 3.2 Generate instructions from the process

- In your process diagram:
- Open ‘Edit’ mode
- In the left panel, click on the ‘magic wand’ icon
- Click on ‘Instructions’
- The Instructions will be displayed in the new window
- Click ‘copy’ to copy all instructions

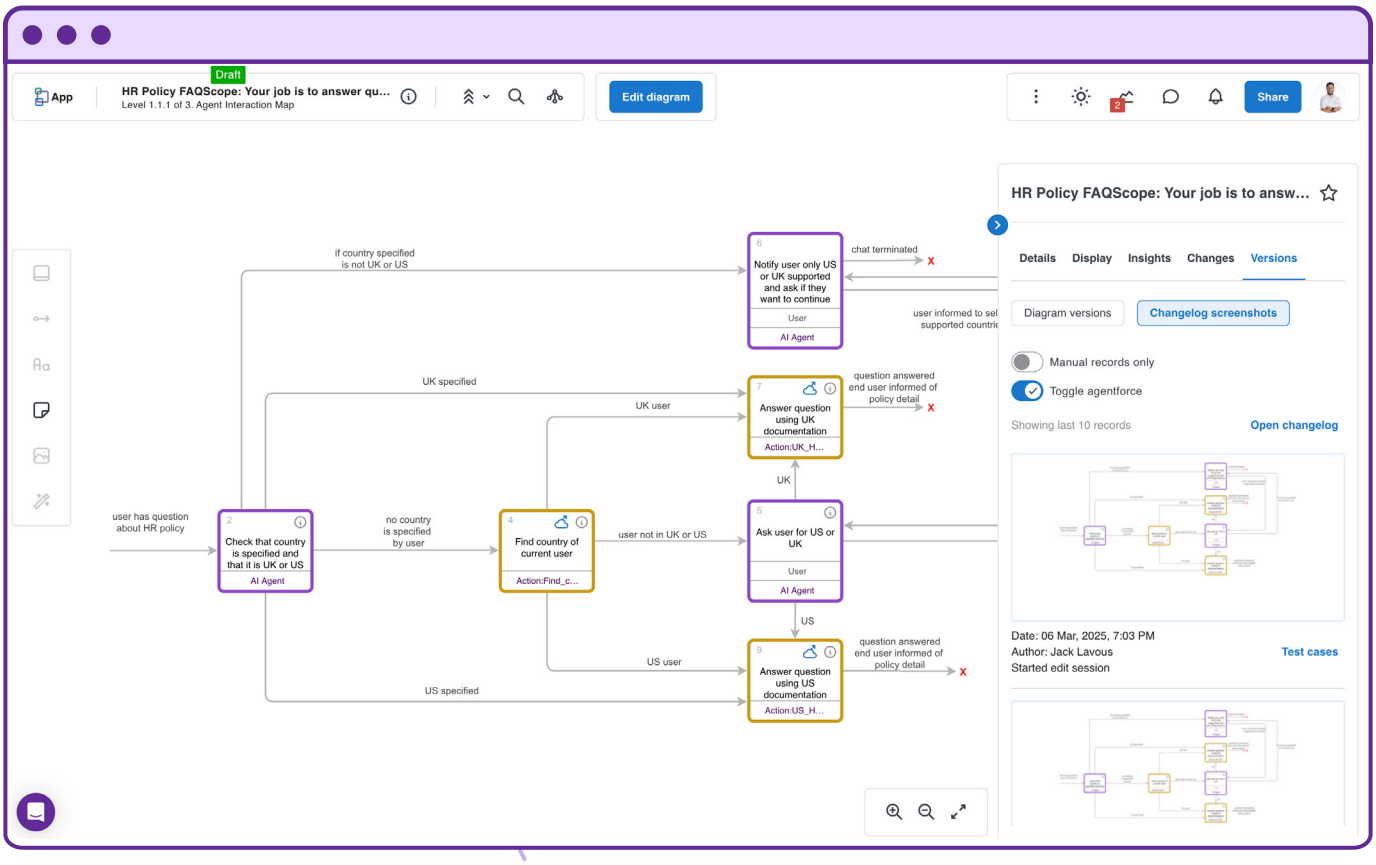



Go to the Agent Builder and the Topic. Paste instructions into ONE instruction field. There is no need to paste each instruction into a separate instruction fields.

**Salesforce best practice is now only to use one instruction field for all the instructions and guardrails.**

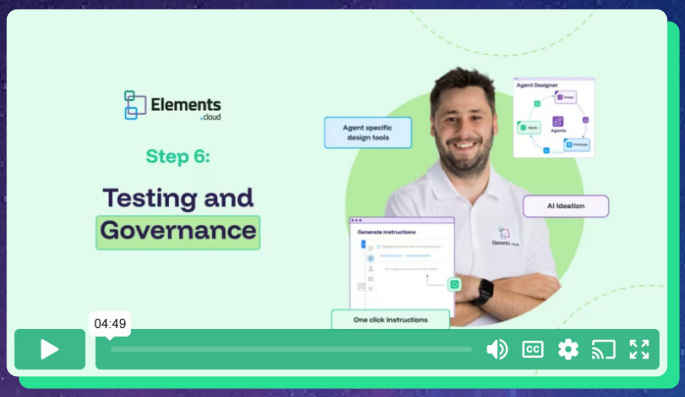
# 3.3 Level 2 AID change log

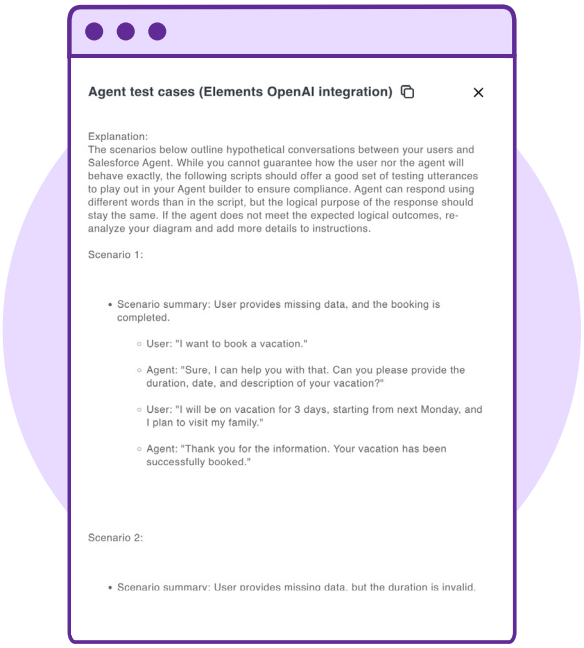
Every time you start to edit the Level 2 AID, it takes a snapshot of the diagram. Also when you generate instructions it takes a snapshot of the diagram and a copy of the instructions. You can access snapshots in the right panel of the diagram. You can filter for just these snapshots.



Step 4

# Evaluate the Topic




Testing involves running each test case systematically to confirm whether the agent behaves as designed. If the agent responds as expected, the test passes. However, if something fails, (i.e. if the agent does not behave as intended), it is critical to address the root cause.

## 4.1 Generate test scenarios

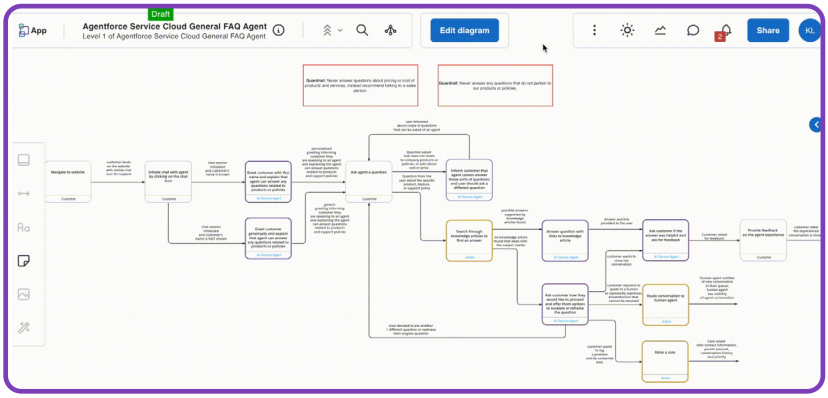
Review the Level 2 AID and look at the different paths and create test scenarios. You wll use these to evaluate the agent. This is all about evaluating the logic and probable outcomes from the multiple conversation permutations that can take place in different scenarios. We are developing significant capabilities to enhance the evaluation of Agents in this context. This document (and users) will be updated as we release this critical capability in the near future.

## 4.2 Generate test utterances

We also generate test utterances for each path to use alongside your analysis of the Level 2 AID.

**In your process diagram:**

- Open ‘Edit’ mode
- In the left panel, click on the ‘magic wand’ icon
- Click on ‘Test cases’
- The test cases will be displayed in the new window
- The test cases will be UAT scripts of conversations between human and Agent based on mapped out process.



## 4.3 Evaluate the Agent

Inside Agent Builder, firstly take your agent through the ‘Happy Path’. Keep your questions and statements clear within the bounds of the instructions you have given it, don’t try and ‘push it’. Take the agent through its activities one at a time, tracking its progress through the process as you go.

Then take the agent through some of the fault paths to evaluate the results.

## 4.4 Fix failures

You need to work out why the agent failed and then how to resolve it. That is made far easier because you have the Level 2 AID:

1. Identify the failing step: Pinpoint where in the Level 2 AID the agent is not responding correctly. This could be due to unclear or insufficiently detailed input text, or activity text. Or you may need to change an instruction to an action.

2. Update the diagram: Make the changes to the diagram based on your analysis of the problem.

3. Regenerate Instructions: Once the diagram is updated, regenerate both the agent instructions and the test utterances.

4. Retest the Topic: Run the updated utterances again to validate the changes.

This iterative process ensures that agent behavior aligns precisely with the designed logic, maintaining clarity and consistency across all steps. And it also helps you keep your design documentation up to date.

Step 5

# Deploy Agent

## 5.1 Deploy Agentforce and related metadata

Once the topic passes all evaluations, you can deploy your Agent using your chosen Salesforce DevOps tool. You may need to get sign-off from stakeholders and the Level 2 AID makes this easier to explain what the Topic does.

**You will need to deploy:**

- The Agentforce metadata (Agent, Topics, Actions)
- Any metadata that you built or updated for the Actions (Apex, Flow, Prompt Templates)
- Any Data Cloud metadata configuration
- Any changes to the Channel where the Agent is deployed - e.g. Experience Cloud

And finally activate the Agent in your Production Org.

## 5.2 Governance of AIM for compliant/regulated use-cases

On top of metadata deployment, use Elements diagram versioning and governance to publish your AIM map. This creates a locked-down version and you can keep working on the "Draft" version of the diagram. (See "Diagram Versions and Publishing new Diagrams" article)

We recommend that the version management is treated as a core deliverable for building Agents. To benefit from it, the AIM content will need to be identified as ongoing operational content that will be handed over to and managed by the end-customer. At the start it may be project or POC stakeholders who are allocated. But even there, "who approved these directions and instructions for the Agent" will be very valuable.

Once in operation, this becomes an operational benefit as the auditable design documentation which is the equivalent of a "Quality Management System" (QMS) for human labor - but applicable to your Digital Labor. The more complex and critical your Agent use cases get, the more valuable this will prove to be for you.

Step 6

# Post Implementation Best Practices

## 6.1 Optimise Agent Performance and Documentation

**ALWAYS** use the diagrams to modify your Instructions, Action Descriptions, Classification and Scope. You can then generate and paste these into Agent Builder. If you do it direct in Agent Builder, you will get out of sync and when you want to build on top of what is in the diagrams, they will not represent the current working agent.

Connect Elements to your Salesforce Org, you can sync the Org metadata including Data Cloud and Agentforce. This provides documentation, impact and dependency analysis

Extend the AIM to add Topics and Agent Instruction Diagrams.
This is your central repository of Agent documentation.

Regularly review and refine agent designs based on performance metrics and feedback.

The dependent metadata in your existing Org to allows you to understand the impact that changes elsewhere in your Salesforce Org will have on the function and performance of your Agent.

***Note:** this is doubly important with Agents since they may continue to function, but simply change their behavior.*

Elements.cloud

# Final Word

By implementing Agentforce with Elements.cloud, you gain a structured, iterative approach to designing, building, testing, and deploying AI agents.

This methodology ensures clear logic, stakeholder approval, and measurable business value while enabling continuous improvement.

Get started for free →