Connect Data Cloud to Agentforce and Prompt Builder

Combine harmonized data and generative AI to create powerful interactions and automations.

Get Started with Data Cloud and AI

Learning Objectives

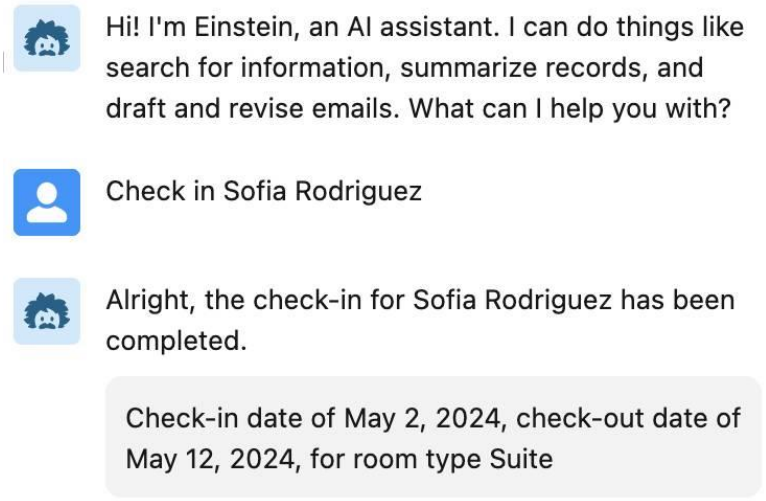
In this project, you'll:

- Sign up for a Data Cloud and Einstein 1–enabled playground.
- Present data stream data to users with a Data Cloud Related List.
- Extend Agentforce with a flow-based action.
- Create a prompt template to use in sales emails.

Welcome to Coral Cloud Resorts

Everyone needs a vacation from time to time. So for your next trip, why not consider Coral Cloud Resorts? Coral Cloud Resorts are known for offering a wide variety of amazing destination activities and doing everything to ensure guests have a fantastic stay. This commitment to excellent service has brought Data Cloud and Salesforce to Coral Cloud's attention. Becca Cloudier, the Salesforce admin, just earned the Salesforce Data Cloud: Quick Look badge, where she learned how Data Cloud can unify data from across many sources. Coincidentally, she's also recently earned the Agents and Agentforce Basics and Prompt Builder Basics badges, discovering just how powerful generative AI can be. Now it's obvious to Becca: Data Cloud and Salesforce go together like sunshine and beaches!

Becca starts making plans for her first project that combines Data Cloud with generative AI: Streamline the check-in process while still keeping it a personalized experience. Resort staff can give more personal attention to their customers, who in turn spend less time standing at a check-in desk.



Once the project is complete, you can witness how a few pieces of data from an external source can weave their way through Salesforce and generative AI functionality, ending in an email that Agentforce drafts in a matter of seconds. So pack your bags, because we're about to take a trip to Coral Cloud Resorts for exciting adventures in Data Cloud and Agentforce!

Sign Up for a Custom Playground with Data Cloud and Einstein AI

To complete this project, you need a custom playground that contains Data Cloud, Einstein AI, and our sample data. If you haven't already clicked the Create Playground button at the top of this page, do that now, and follow the steps to create a custom playground and connect it to Trailhead.

Note: This Custom Playground is designed to work with the challenges in this badge, and may not work for other badges. Always check that you're using the Trailhead Playground or special Developer Edition org that we recommend.

Also note that your developer org has a custom permission set that doesn't have access to Data Cloud Setup. It's intended to be used only for specified Trailhead challenges.

Bring External Data into the Contact Page Layout

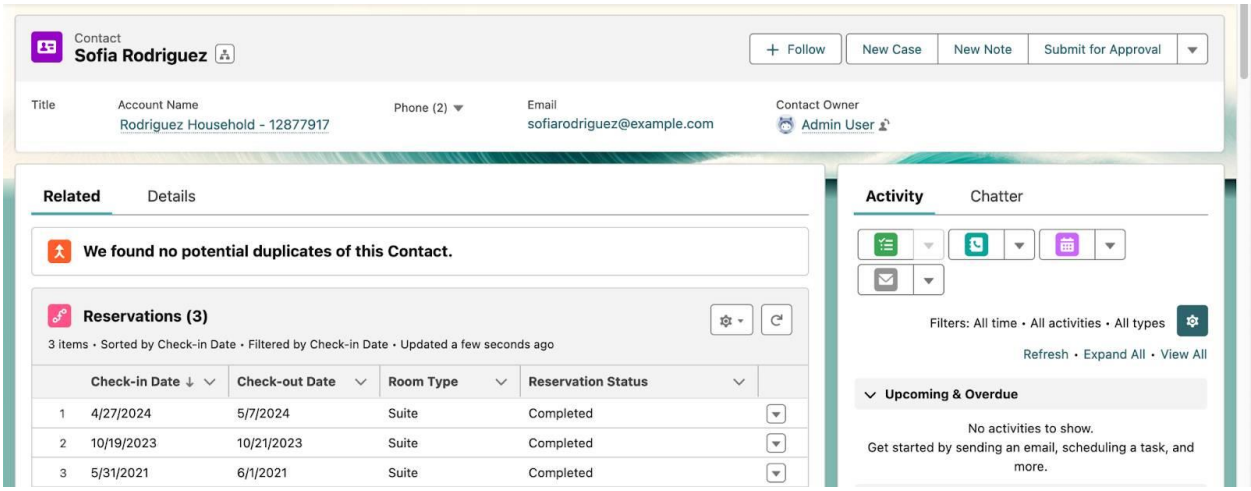
Make Salesforce an All-Inclusive Resort, for Data

Becca is excited to get working on her first Data Cloud plus AI combo project. But before starting, she knows it's important to take stock of the current systems and processes. Streamlining check-ins involves Coral Cloud's current reservation system, a slightly antiquated solution called Reserv-o-matic. This external system allows guests to place many reservations under a single profile.

Each guest has several fields to hold information like their name, email addresses, and phone number. One such guest, Sofia Rodriguez, has created a few reservations, each with check-in and check-out dates, along with room type and reservation status.

At present, all of that information is completely separate from Coral Cloud's Salesforce org. In that org is a Contact record for Sofia, with tons of existing

related data. Data Cloud will make the connection between the Sofia in Salesforce with the Sofia in Reserv-o-matic. And once that relationship is established, Becca can do all sorts of interesting things with it. For example, she can start by putting a reservation related list right into the Contact page layout.



To accomplish this, Becca has to complete three steps.

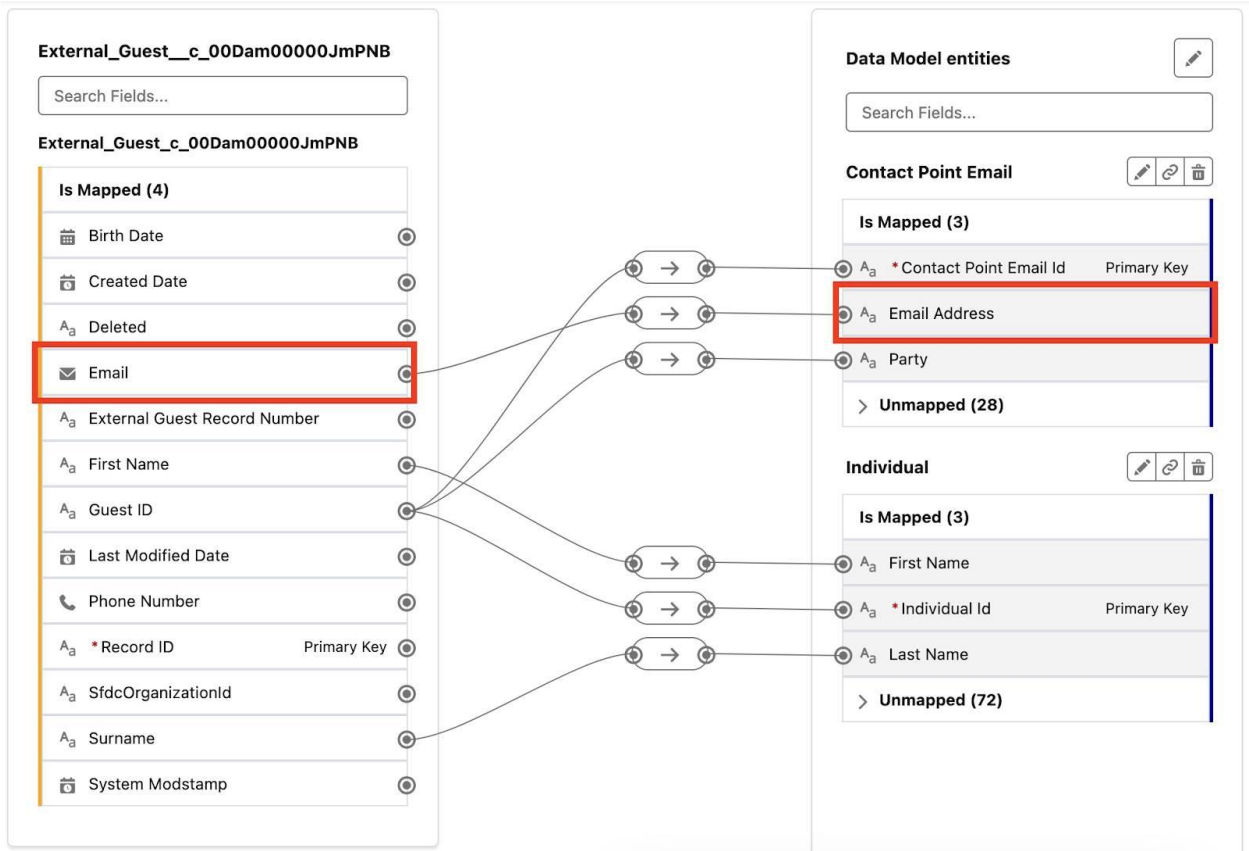
1. Make Data Cloud aware of Reserv-o-matic data.
2. Find matches between external guests and internal contacts.
3. Update the Contact page to display external reservations.

With the plan in place, it's almost time to begin setting things up. But a quick note first, intrepid Trailblazer. As you may have guessed, Reserv-o-matic doesn't actually exist. So that means there's no external data source for you to use in this project. For that reason, Team Trailhead has put the "external" guest and "external" reservation data into custom objects that reside in your special org. So throughout this project we're going to *pretend* the data exists outside of your org, and you get to play along. Make sense? OK, back to our regularly scheduled programming.

Enable Data Streams for Existing Coral Cloud Data

At this point in our story, Team Trailhead has saved a little time for you by already creating a connection between Data Cloud and Reserv-o-matic. However, Data Cloud is not yet aware of what data exists in the external system, let alone how it relates to internal objects. So the first step Becca must take is to identify the data that should be brought into Data Cloud. That means creating a **data stream**, a fundamental part of Data Cloud administration.

Data streams describe where to find data from within a given connection. Each data stream creates a related **Data Lake Object (DLO)**, which is a storage container for the data coming from the data stream source. But how does that new DLO relate to all the other data that already exists in Data Cloud? The answer comes in the form of **Data Model Objects (DMOs)**, which describe how data is structured, sort of like metadata. For example, the DMO named Contact Point Email has details about how to properly store an email address, regardless of where it comes from. So Becca can map the email column from the external guest data to the email address field in the DMO.



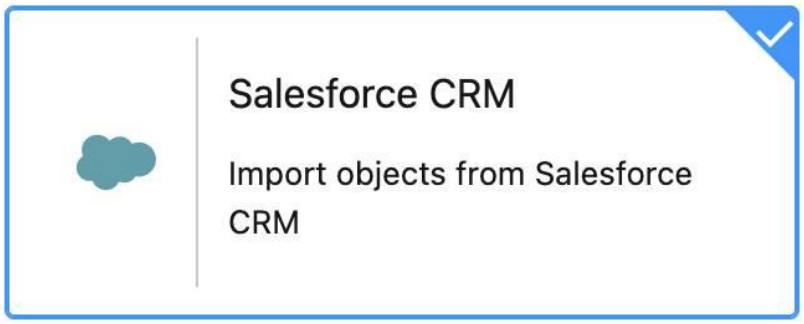
That mapping makes it possible for Data Cloud to use data in clever ways. In this project, Data Cloud uses the email address to know that Sofia from Reserv-o-matic is the same Sofia in Salesforce.

Creating data streams and mapping them is a whole topic on its own, which you can learn more about in the Create a Data Stream in Data Cloud project. In this project, we've simplified things by packaging all of the Coral Cloud data streams and mappings into a data kit, which has already been loaded into your special org. All you have to do is enable the data streams. You start by launching your Data Cloud and Einstein-enabled playground.

Enable Data Streams

1. If you haven't already, launch your Data Cloud and Einstein 1 playground.

2. Click the **App Launcher** , type data, then click the **Data Cloud** app.
3. Click the **Data Streams** tab.
4. Click **New**.
   This is the point where you have to play pretend, and choose an internal source of data instead of an external one.
5. Click **Salesforce CRM**.



6. Click **Next**.

7. Click the **AIPlusData** Custom Data Bundle.

8. Click **Next**.
9. Notice that the default Data Space is pre-selected and all the fields included in the bundle are listed. Leave the selections as is, and click **Next**.
10. Click **Deploy**.

Well done, you've identified important sources of data by enabling data streams. Over the next few minutes the data is ingested into the Data Lake Objects. And since the Data Model Objects and mappings already exist, you can soon make use of the data.

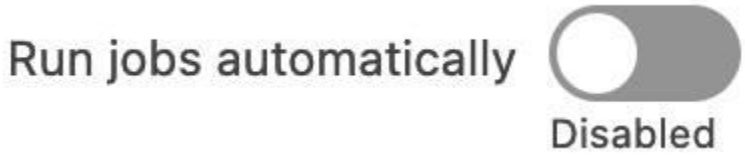
Set Up Identity Resolution

Becca knows that many of the guests from Reserv-o-matic are the same people that have Contact records in Salesforce. Since both guest and contact are mapped to common Data Model Objects, she can use a powerful feature of Data Cloud to match the Sofia that's in Salesforce with the Sofia from Reserv-o-matic for one unified Sofia. It's called identity resolution, and it's key to bridging the gap between Salesforce Contacts and the external reservation data. You start by navigating to the Identity Resolutions tab to create an Identity Resolution ruleset.

1. Click the **Identity Resolutions** tab.
2. Click **New**.
3. Click **Create New Ruleset**, then click **Next**.
4. Leave **default** selected in the Select Data Space menu.
5. For Primary DMO, choose **Individual**.
6. Type `ccid` for Ruleset ID.

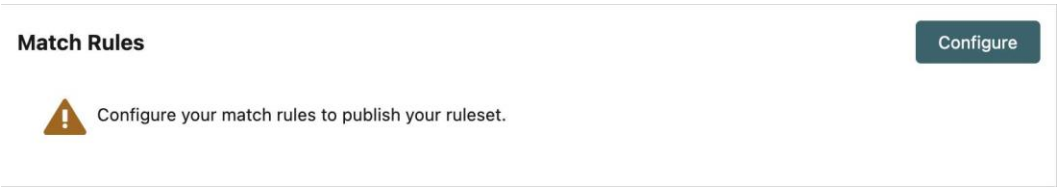
   This stands for Coral Cloud Identification.
7. Click **Next**.
8. Type `Guest Name and Email` for Ruleset Name.

9. In the Description field, type `Reconcile Reserve-o-matic contact data with Salesforce contact data`.
10. Disable **Run jobs automatically**.



You will manually start a job later.

11. Click **Save**.
    The framework of the ruleset is saved, now it's time to set up how matching is done.
12. Click **Configure**.



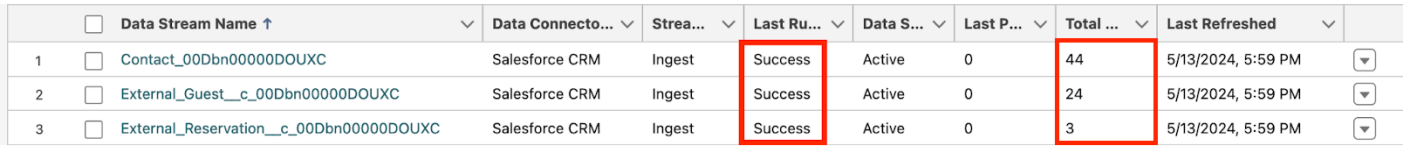
13. Click **Next** (or pause here and read through the Match Rule Instructions.)
14. Click **Configure**.
15. Choose **Fuzzy Name and Normalized Email**.
    This determines the primary pieces of data that are used for matching.
16. Click **Next**.
    There are quite a few sophisticated ways to fine-tune how matches are made, but for now you'll just use the defaults.
17. Click **Next**.
18. Click **Save**.

Excellent, now Data Cloud knows how to detect when two individuals are actually the same person, using a ruleset. Your org has a few dozen guests and contacts that will match when using this ruleset. To learn more about identity resolution rulesets, check out the Quick Start: Create an Identity Resolution Ruleset project.

Verify Data Is Ingested, and Run an Identity Resolution Ruleset

Rulesets are all about matching data, but you need that data to be present in your Data Lake Objects before any matching can be done. Let's verify that the data streams have finished ingesting all the sample records before running the ruleset.

1. Click the **Data Streams** tab.
2. Refresh the page in your browser.
   All streams need to state **Success** for Last Run Status and a number (not zero) needs to be present in the Total Unified Profiles column. If you don't see these, click the dropdown arrow on the right side of each data stream and select **Refresh Now**. You may need to refresh your browser again, too.



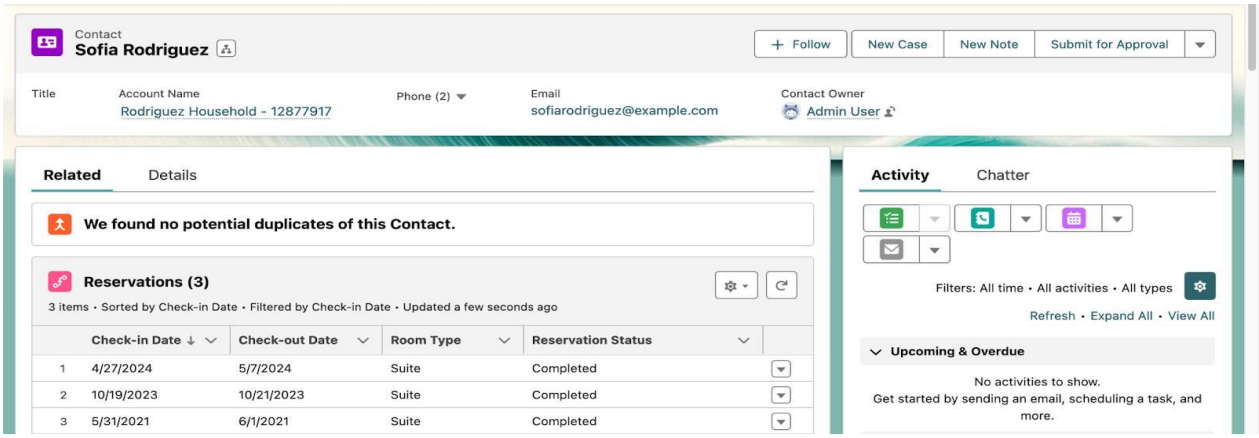
	Data Stream Name ↑	Data Connecto...	Strea...	Last Ru...	Data S...	Last P...	Total ...	Last Refreshed
1	Contact_00Dbn00000DOUXC	Salesforce CRM	Ingest	Success	Active	0	44	5/13/2024, 5:59 PM
2	External_Guest__c_00Dbn00000DOUXC	Salesforce CRM	Ingest	Success	Active	0	24	5/13/2024, 5:59 PM
3	External_Reservation__c_00Dbn00000DOUXC	Salesforce CRM	Ingest	Success	Active	0	3	5/13/2024, 5:59 PM

3. Click the **Identity Resolutions** tab.
4. Click **Guest Name and Email**.
5. Click **Run Ruleset** in the upper right corner.

Sometimes it takes a little while for a ruleset to run. It should be pretty quick for you because the org only has a few dozen records, but even so you can do other things while it processes. Like creating a Data Cloud Related List! Let's do that next.

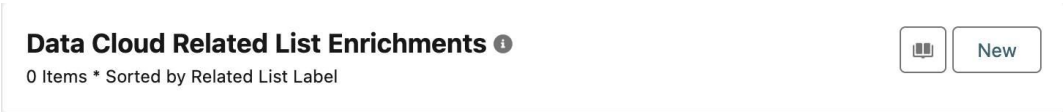
Connect Data Cloud and CRM with Enrichment

One awesome benefit of having all of your data accessible in one place is that you can create experiences that seamlessly blend data from all sources. In the case of Coral Cloud, Becca wants to present external reservation data in the same place that hotel staff finds other relevant information about their guests: the Contacts page.



Without knowing better, it would be easy to think this screenshot shows a normal related list–it fits right in! But it's actually a special kind of related list called a **Data Cloud Related List**. Creating one is done under Setup. (A lot of popup windows appear as you complete the remaining steps. Feel free to dismiss them.)

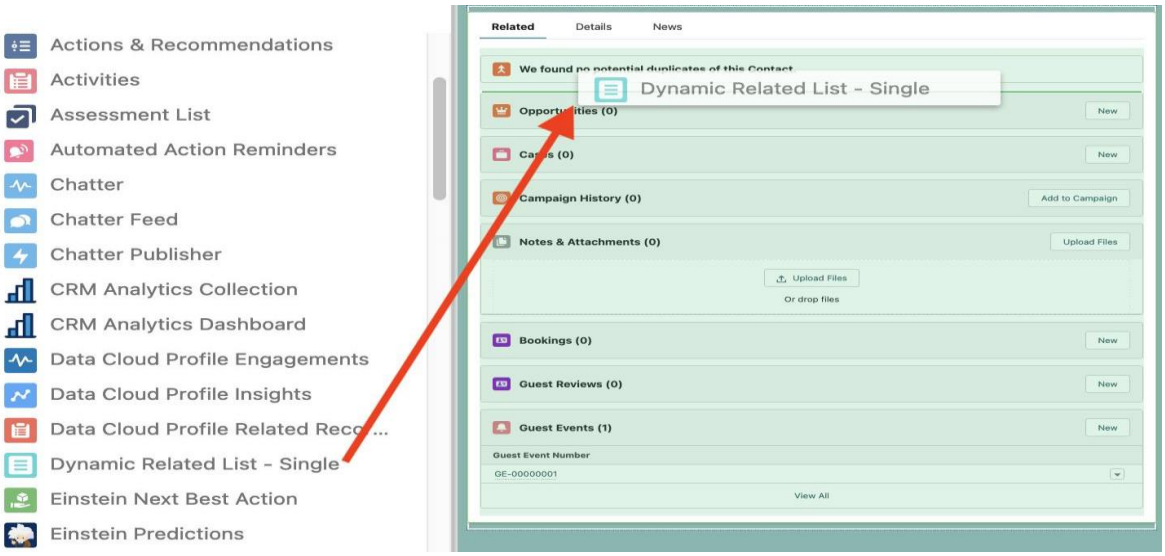
1. Click the **Setup** icon , then click **Setup**.
2. Click **Object Manager**.
3. Click **Contact**, (not Contract).
4. Click **Data Cloud Related List**.
5. Click **New**.



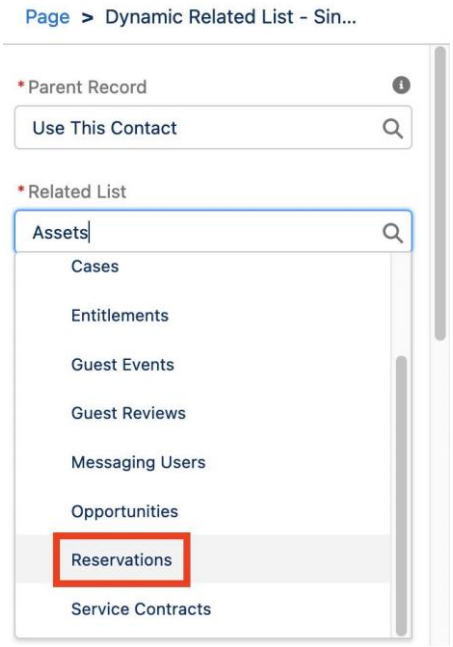
6. For Data Cloud Object, choose **ExternalReservation**.
7. Click **Next**.
8. If you're using a newly created Trailhead Playground org, skip this step. If you're using an older org that doesn't have Enhanced Security Data Spaces, check **Enable Controlled by Parent Permissions**, and then click **Next**.
9. Update Child Relationship Name to Reservations.
10. Click **Next**.
11. Update Related list label to Reservations.
12. Click **Next**.

OK, the Data Cloud Related List exists, but it's not yet on any page layouts. So Becca updates the Lightning record page for Contact, and configures how it appears.

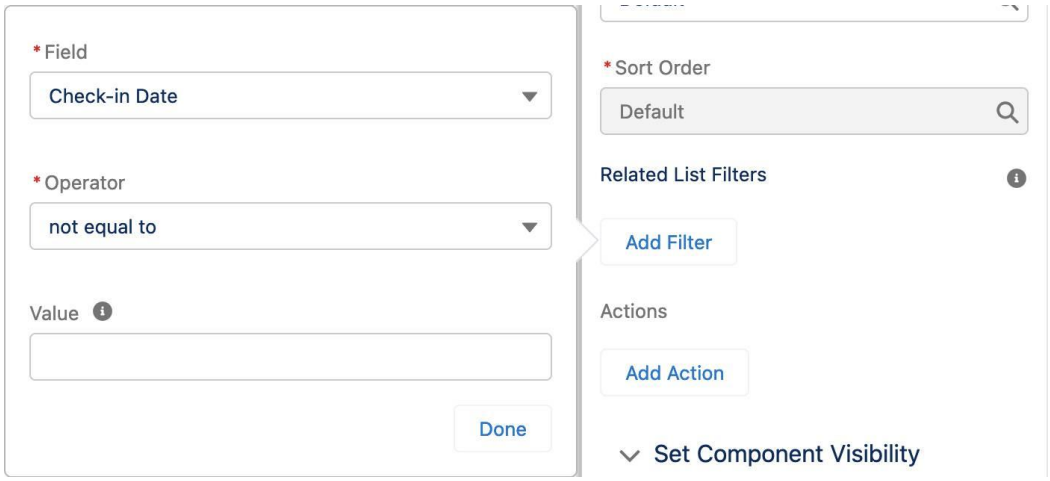
1. In the left menu, click **Lightning Record Pages**.
2. Click **Contact Record Page**.
3. Click **Edit**.
4. Drag a **Dynamic Related List - Single** into the Related tab above Opportunities.



5. In the right menu, under Related List, choose **Reservations**.



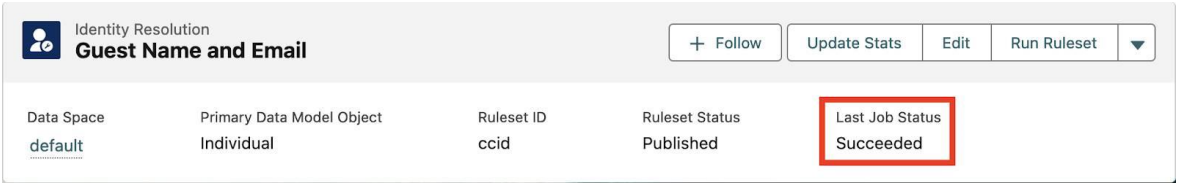
6. Under Related List Fields, click **Add Field**.
7. Find and select **Check-in Date**.
8. Click **Done**.
9. Repeat steps **6-8** for **Check-out Date**, **Room Type**, and **Reservation Status**.
10. Delete the **Record ID** field at the top of the list by clicking the **X**.
11. Under Related List filters, click **Add Filter**.
12. For Field, choose **Check-in Date**.
13. For Operator, choose **not equal to**.
    In other words, if the check-in date isn't empty, show the reservation. If a filter isn't added, only the past 7 days of data appear. In your org the data is probably older than that.



14. Click **Done**.
15. Click **Save**.
16. Close the Lightning App Builder tab to return to the Data Cloud app.

Excellent, the Data Cloud Related List is ready to go! Let's check out Sofia's Contact record to verify that her reservations are listed.

1. Click the **Identity Resolutions** tab, then click the **Guest Name and Email** ruleset. Refresh your browser and verify that the Last Job Status field states Succeeded. If not, wait a minute and refresh again.



2. Click the **App Launcher**, type sales, then click the **Sales** app.
3. Click the **Contacts** tab.
4. Click **List View** (on the far right), then select **All Contacts**.
5. In the **Search this list…** box, type Sofia Rodriguez and click enter.
6. Click **Sofia Rodriguez**.

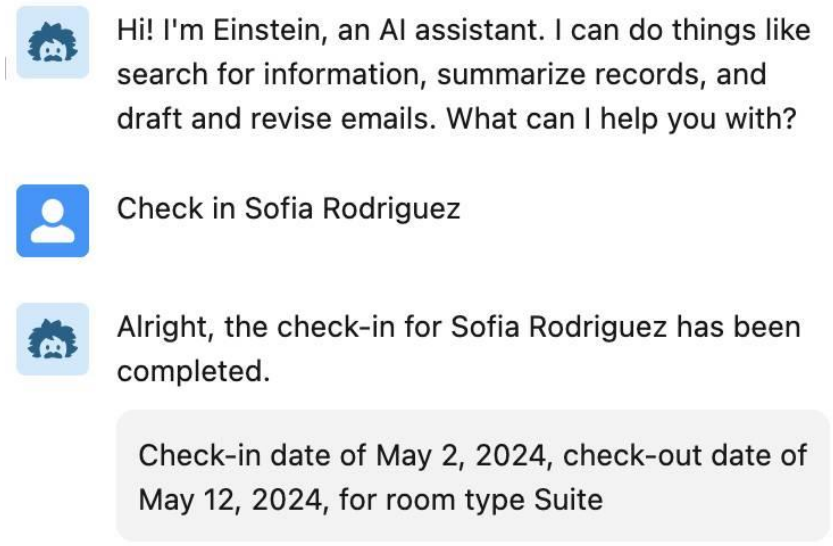
Sofia should have three reservations, now appearing right inside of the contact page layout.

Enhance Agentforce to Act on Data with Conversational Language

Give the Warmest of Welcomes

Every day at Coral Cloud Resorts, guests arrive having traveled from around the globe, and they're usually ready to start relaxing. But before they can, they have to check in with the front desk. So Becca is going to make that last little hurdle as effortless as possible for her staff so that guests can begin vacationing ASAP.

Whenever a check-in occurs, Coral Cloud staff logs the event by creating a record on a custom object named Guest Event. They use it to track a lot of things, like when guests request housekeeping or when guests attend a social activity. Right now staff must manually create the check-in record, but Becca sees two ways to level up the process. First, she uses a flow to create a Guest Event record based on the latest reservation data in Data Cloud. Second, she teaches Agentforce how to launch the flow through conversational language. So when Sofia Rodriguez arrives to begin her stay, whoever is at the front desk can simply ask Agentforce to "check in Sofia Rodriguez" and Agentforce does the rest!



Finish a Flow to Log Guest Events

Thanks to Data Cloud, Becca has all the information required to make a flow that starts with a contact, like Sofia Rodriguez, and creates a record based on reservation data. However, the path between the two is a little twisty. As you learned in the previous step, identity resolution automatically creates bridging objects that connect related Data Model Objects. In order to make the check-in flow, Becca has to cross one of those bridges, twice.

The bridge in question is named **Unified Link Individual ccid,** which connects the Sofia in Salesforce to the Sofia in Reserv-o-matic. Here's a simplified version of the data in that bridge.

Full Name	Data Source	Source ID	Unified ID
Sofia Rodriguez	Salesforce	003bn000002FILwAAO **[1]**	ca6369777b7dbca79215f46016e74dd6
Sofi Rodriguez	Reserv-o-matic	10008155 **[2]**	ca6369777b7dbca79215f46016e74dd6

Notice that the Unified ID is the same for both Sofias. So if you know Sofia's Salesforce ID [1], you can find her Reserv-o-matic ID [2] by way of the Unified

ID. That's important because reservations are only tied to guests by the Reserv-o-matic ID. Here's a simplified version of that.

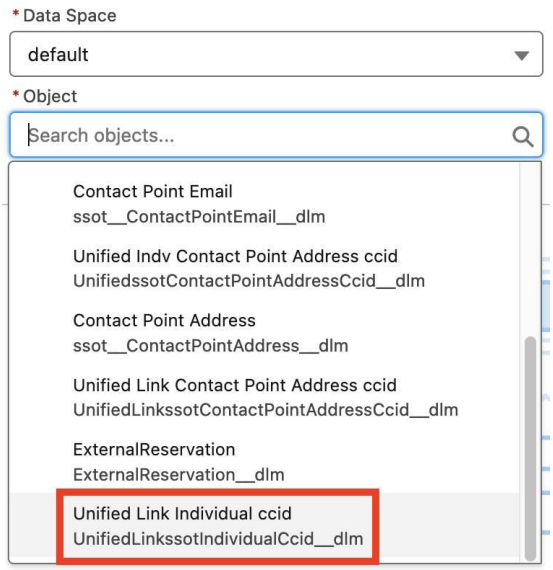
Contact ID	Check-in Date	Check-out Date	Room Type	Reservation Status
10008155 **[2]**	5/31/2021	6/1/2021	Suite	Completed
10008155 **[2]**	10/19/2023	10/21/2023	Suite	Completed
10008155 **[2]**	4/27/2024	5/7/2024	Suite	Completed

With all of that in mind, let's break down what the flow does.

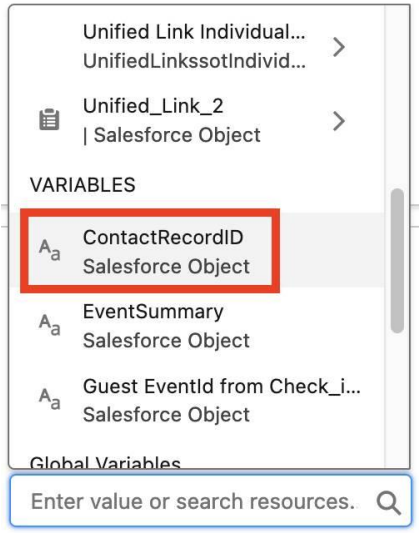
- Knowing the Source ID from Salesforce [1], query the bridge to find the Unified ID.
- Knowing the Unified ID, query the bridge again to find the Reserv-o-matic ID [2].
- Knowing the Reserv-o-matic ID [2], query the reservation data to find the latest reservation details.
- Knowing the latest reservation details, update a flow variable with the details.
- Create a Guest Event record for check-in using the variable.

It sounds like a lot, but we have good news for you: Team Trailhead has already created most of the flow. The remaining parts involve objects that only exist now that you've created data streams and set up identity resolution. So let's finish that flow!

1. Click the **Setup** icon, then click **Setup**.
2. Type flows in the search, then click **Flows** (under Process Automation).
3. Click **Create Check-in Guest Event**.
4. Click the **Unified Link 1** element, then click **Edit Element**.
5. For Data Source, choose **Data Cloud Object**.
6. For Data Space, choose **default**.
7. For Object, choose **Unified Link Individual ccid**.
   Be sure to choose the right object, there are a few that have similar names.



8. In the Filter Records section, for Field, choose **SourceRecordId__c**. This is the actual name of the Source ID column in the first table.
9. For Value, choose the variable **ContactRecordId**. ContactRecordId is a variable Team Trailhead created in advance. It just represents the Salesforce ID of the contact record.



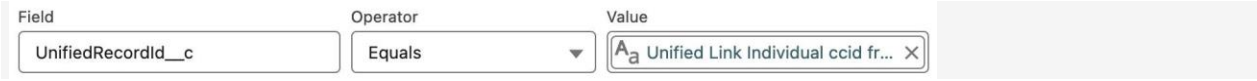
10. Click **Save**.

OK, the flow has pinpointed the first record in the Unified Link table. Time to use the Unified ID from that record to find the Reserv-o-matic ID.

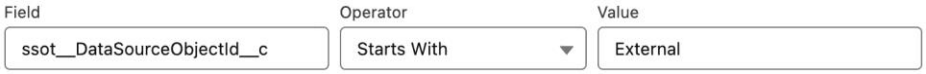
1. Click the **Unified Link 2** element, then click **Edit Element**.

2. For Data Source, choose **Data Cloud Object**.
3. For Data Space, choose **default**.
4. For Object, choose **Unified Link Individual ccid**.
   This is in fact the same object from the first element.
5. In the Filter Records section, for Field, choose **UnifiedRecordId__c**.
   This is the actual name of the Unified ID column from the first table.
6. For Value, copy and paste `{!Unified_Link_1.UnifiedRecordId__c}`.

   This is a pointer to the Unified ID from the first query. When you click away from the Value, the text is replaced with a placeholder.



7. Click **Add Condition**.
8. For Field, choose **ssot__DataSourceObjectId__c**.
9. For Operator, choose **Starts With**.
10. For Value, type `External`.



The Data Lake Object is really called ExternalReservation, not Reserv-o-matic like in the simplified table. By adding this extra condition the flow will ignore Sofia's Salesforce ID and find 10008155 from the first table.

11. Click **Save**.

Now that you know Sofia's Reserv-o-matic ID, you can find her most recent reservation.

1. Click the **Reservation 1** element, then click **Edit Element**.
2. For Data Source, choose **Data Cloud Object**.
3. For Data Space, choose **default**.
4. For Object, choose **ExternalReservation**.
5. In the Filter Records section, for Field choose **Contact_ID_c__c**.
6. For Value, copy and paste `{!Unified_Link_2.SourceRecordId__c}`.

   This is a pointer to the previous query.

7. For Sort Order, choose **Descending**.
8. For Sort By, choose **Check_in_Date_c__c**.
9. Click **Save**.

Sorting in this way puts the most recent reservation at the top of the results, so it's used for the rest of the flow. The final piece of the puzzle is to use the reservation details to set the value of a variable (EventSummary), which is ultimately placed into a new Guest Event record.

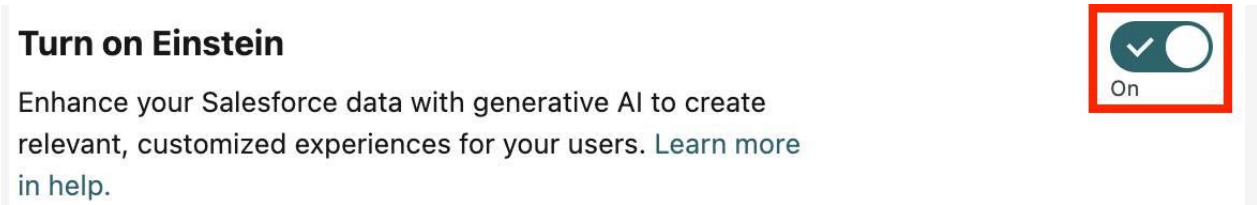
1. Click the **Event Summary** element, then click **Edit Element**.
2. For Value, copy and paste the following:
   `Check-in date of {!Reservation_1.Check_in_Date_c__c}, check-out date of {!Reservation_1.Check_out_Date_c__c}, for room type {!Reservation_1.Room_type_c__c}`.
3. Click outside the Value box so the Save button becomes available, then click **Save**.
4. Click **Activate**.
5. Close the Flow tab to return to the Setup tab.

Great job shaping up the flow. It was a bit of work, but it'll pay off almost immediately once Agentforce knows how to use it.

Create a Check-in Agentforce Action

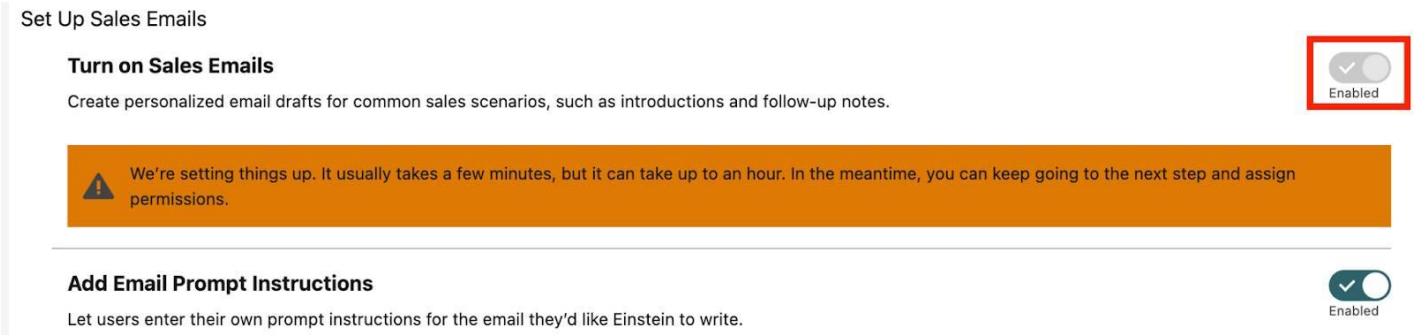
There's more good news for you: setting up Agentforce to use a flow is super easy.

1. Click the **Setup** icon  , then click **Setup**.
2. In the Quick Find box, type `Einstein Setup`, then click **Einstein Setup**.
3. Toggle the Turn on Einstein switch to **On**.

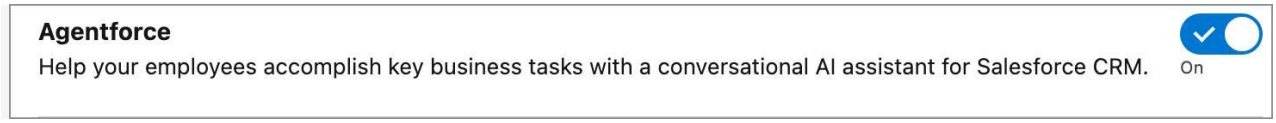


4. In the Quick Find box, type `Einstein Sales`, then click **Einstein for Sales**.

5. Sales Emails may already be enabled in your org, but if not, toggle the Turn on Sales Emails switch to **Enabled**. It may take some time to set things up, but you can continue on in the meantime. After a few minutes, you may need to refresh your browser to clear the setup alert.

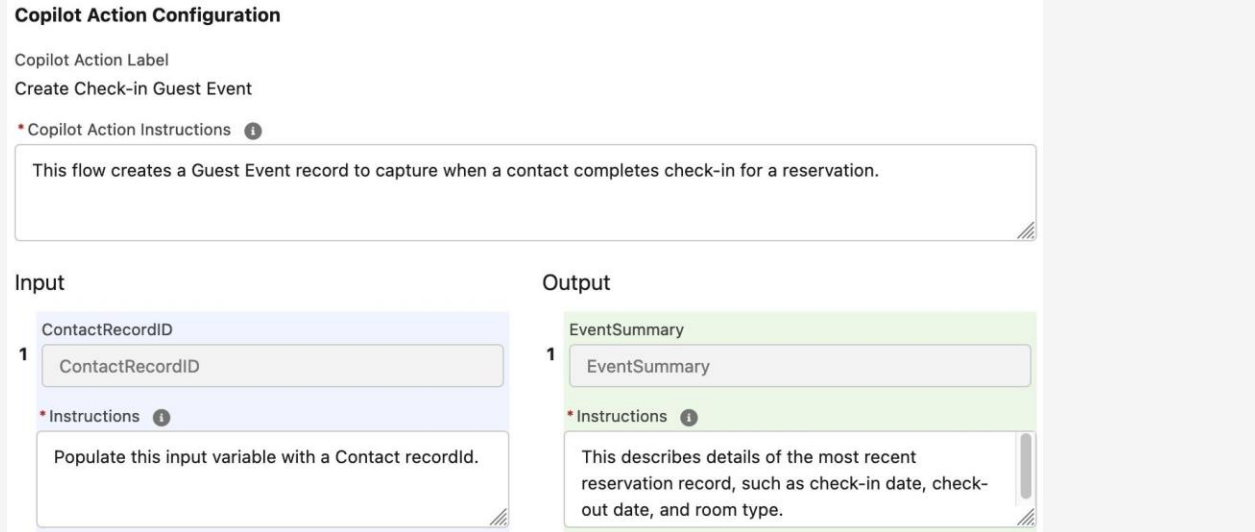


6. In the Quick Find box, type agent, and then click **Agents** under Agent Studio. If you can't find the Agents page, refresh the browser and try again.
7. Toggle the Agentforce switch to **On**.



1. Refresh the browser.
2. In the Quick Find box, type agent, then click **Agent Actions**.
3. Click **+ New Agent Action**.
4. For Reference Action Type, choose **Flow**.
5. For Reference Action, choose **Create Check-in Guest Event**.
6. Click **Next**.

Let's take a moment to look at what Agentforce brings in from the flow. Most importantly, the flow has a well-defined description, which is placed into the Agentforce Action Instructions. The variables have good descriptions too, placed into the input and output instructions. These are critical for Agentforce to understand the purpose of the flow, and how it should be used. Kudos to Becca for following best practices and documenting her work well!

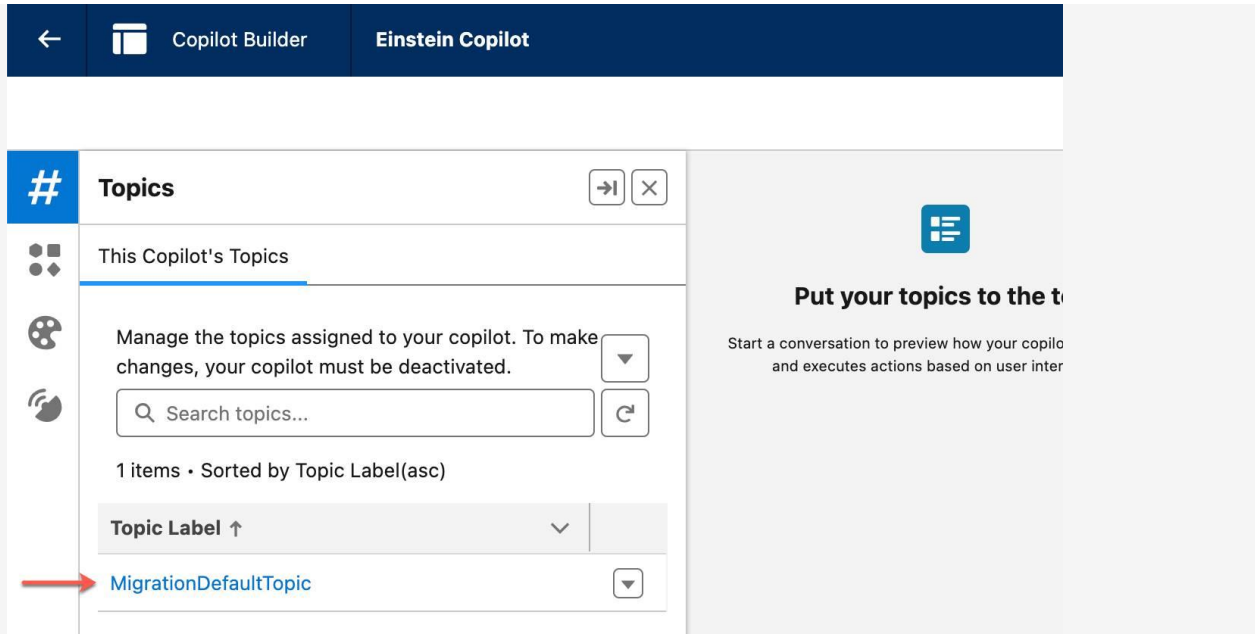


14. Deselect **Show loading text for this action**.
15. Under Input, check **Require input**.
16. Under Output, check **Show in conversation**.
17. Click **Finish**.

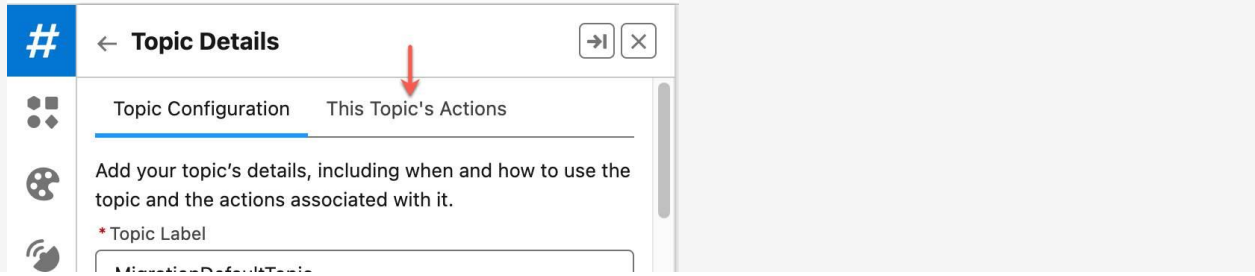
    Your new action is finished, but we need to add it to Agentforce.
18. With agent still in the Quick Find box, click **Agents**.
19. Click **Einstein Copilot**.

20. Click **Open in Builder**.

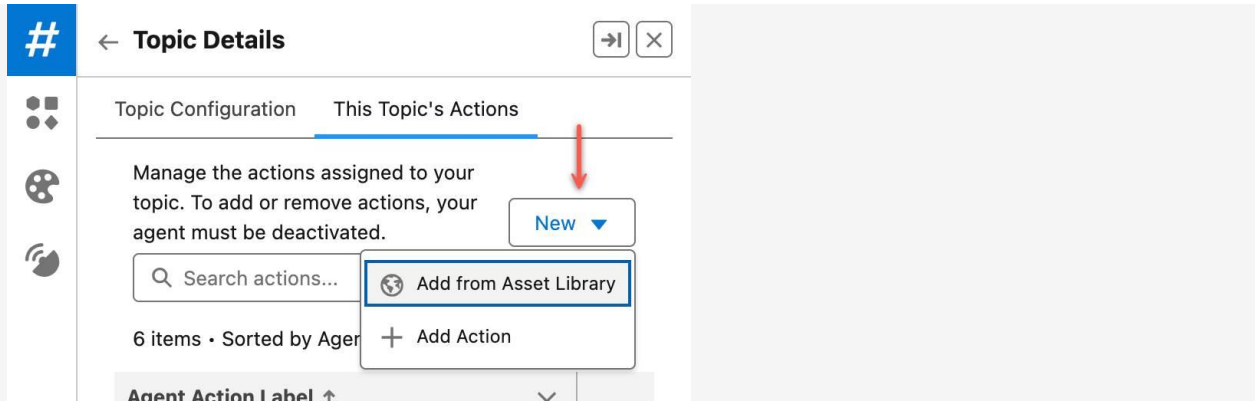
    Ensure your assistant isn't active, it should show an Activate button. If it shows a Deactivate button, click Deactivate.
21. From Topics, click **MigrationDefaultTopic**.



22. Click **This Topic's Actions**.



23. Click **New**, then select **Add from Asset Library**.



24. Check the **Create Check-in Guest Event** box and click **Finish**.

25. Click **Activate**.
26. Close the Agent Builder tab to return to the Sales app.

That's it, Agentforce is ready to help check in guests. The only thing left is to try it out. You may already have the Sofia Rodriguez contact open, so navigate to the Home tab as though you're starting from the beginning of check-in.

1. In the Sales app, click the **Home** tab.
2. Refresh the browser window.
3. Click the **Agentforce** icon to open the Agentforce panel.

4. In the Agentforce panel, type `Tell me about the Sofia Rodriguez contact`, then press enter.This step isn't strictly necessary, but it illustrates Agentforce's ability to find and summarize a record.
5. Click the link for **Sofia Rodriguez**.
   This brings you to Sofia's contact record. If Agentforce didn't reply with a link to the record (which sometimes happens in our training environment), just navigate to her contact record.
7. In the Agentforce panel, type `Check in Sofia Rodriguez`, then press enter.
   After a few moments, the Agentforce agent should respond that check-in is complete, along with some details of the check-in. The Guest Event record was created too.
8. Scroll down to find the Guest Events related list, then click the new **GE record**.

Great, the check-in event is captured just as expected.

9. Close the Agentforce panel.

Phase 2 of Becca's plan is a success. The resort staff loves how easy it is to check-in guests, and the social coordinator is already asking Becca to teach Agentforce how to create Guest Events for his activities.

Add Data to a Prompt Template to Quickly Draft Emails

Create and Use a Sales Email Prompt Template

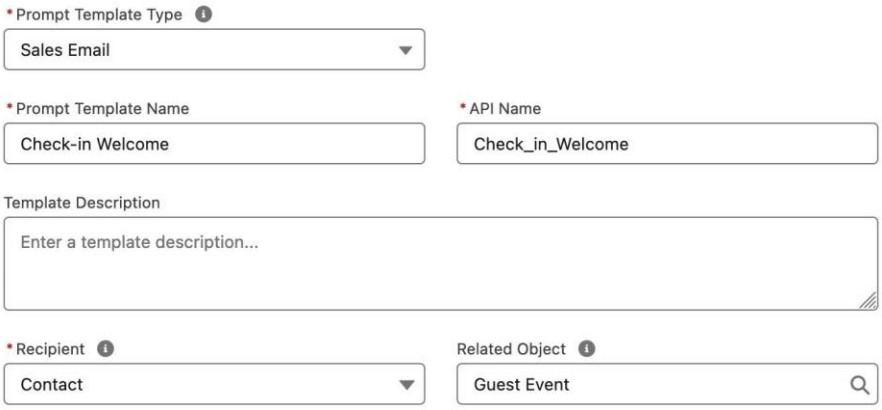
Coral Cloud goes to great lengths to give its guests a fantastic vacationing experience that feels perfectly tailored to their interests. That level of personalization takes a lot of effort. One example is the welcome email that every guest receives upon check-in. It includes more than just the WiFi password. Staff at the front desk often add small details from the conversations they have during check-in. This time-consuming task is about to get a lot easier thanks to Prompt Builder.

As you learn in the Prompt Builder Basics badge, the key to scalable generative AI is to use prompt templates to tell a large language model (LLM) how it should generate content. Prompt templates merge specific data, such as reservation details, with general instructions to make a unique prompt each time the template is called. There's an art to creating prompt templates, which you can learn about in the Prompt Fundamentals badge. And if you want to know more about LLMs in general, check out the Large Language Models badge.

Becca wants to create a prompt template for drafting welcome emails. She wants to include details about the most recent reservation and allow front desk staff to quickly add personal touches to the generated messages. She starts by navigating to Prompt Builder in Setup.

1. Click the **Setup** icon  , then click **Setup**.
2. Type `prompt` in search, then click **Prompt Builder**.
3. Click **New Prompt Template**.
4. For Prompt Template Type, choose **Sales Email**.
5. For Prompt Template Name, enter `Check-in Welcome`.

6. Keep `Check_in_Welcome` as the API name.
7. For Description, type `Check-in welcome email`.
8. For Related Object, choose **Guest Event**.



9. Click **Next**.
   For the sake of time, instead of writing a prompt from scratch, let's just use the one Becca has been working on.
10. Paste the following into the Prompt Template Workspace.

```
You're a receptionist for the Coral Clouds Resort hotel.
Your name is {!$Input:Sender.Name}, with phone {!$Input:Sender.Phone} and
email {!$Input:Sender.Email}.
A guest has just checked-in, and you want to send them a welcome email.
Instructions:
'''
Use clear, concise, and straightforward language using the active voice and
strictly avoiding the use of filler words and phrases and redundant language.
Generate a subject line that can increase open rate using words and content
that is related to the email body content.
Generate the body of the email, which will have the next format:
- First add a title that includes the name of the guest and welcomes them:
{!$Input:Recipient.Name}.
- Add a paragraph explaining the characteristics of the reservation described
in {!$Input:Guest_Event__c.Details__c}, including its check in date, check
out date, and room type.
- Explain breakfast is available from 6 to 10 AM. Find in the reservation if
it's included and add this information to the email, mentioning it can be
included any time.
- Add the wifi network network: Coral_Cloud_Guests. Mention for password,
guests will be prompted for their surname and room number.
- End the email by encouraging the guest to contact you if they have any
inquiries.
'''
```

```
Now generate the welcome email to your guest.
```

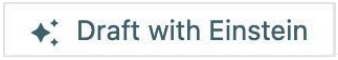
9. Click **Save**.
10. Click **Activate**.

Notice that Becca's prompt includes multiple merge fields, which are colored blue. One field, **Guest_Event__c.Details__c**, contains data that originated from Reserv-o-matic. Nice for it to make an appearance!

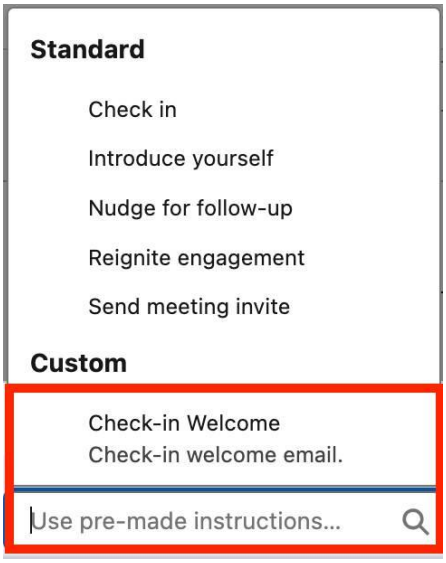
Test Your Work

The final step for any good admin is to test their work, and that's exactly what Becca does next.

1. Close the Setup tab to return to the Sales app.
2. Click the **Contacts** tab.
3. Click **Sofia Rodriguez**.
4. Click the **Email** icon.

5. Click **Draft with Einstein** 

.
6. Click **Use pre-made instructions**, then choose **Check-in Welcome**.



7. For Guest Event, choose the **GE record** created in the previous section.
8. Click **Continue**.
   At this point Agentforce is writing the welcome email based on the new prompt template, and in a moment the draft will appear. Front desk staff can review the results, and add in any personal touches they'd like.
9. Click **Done**.
10. Click **Send**.

And there you have it. In just a few clicks, a personalized and informative welcome email is on its way to another happy guest. Becca is quite satisfied with the way her project combined Data Cloud, Agentforce, and Prompt Builder to improve the experience of guests and staff alike. And after a short moment of reflecting on the success of the project, Becca sets out to find the social coordinator to begin their next AI + Data adventure.