

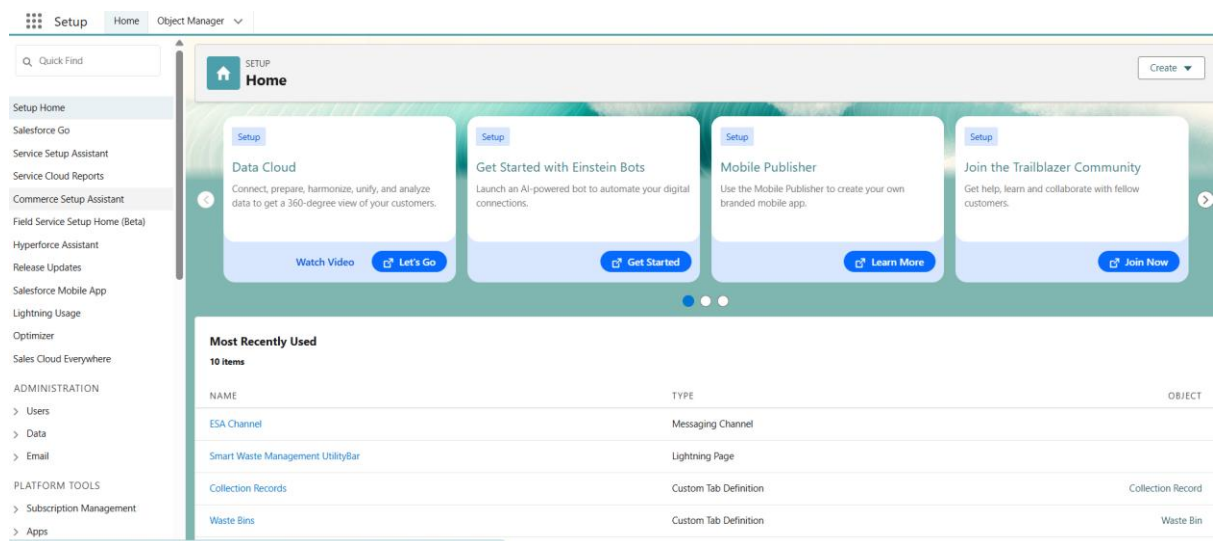
Smart Waste Management Tracker – Salesforce Project

Phase 4: Automation – Record-Triggered Flow

Step 1: Goal of Phase 4

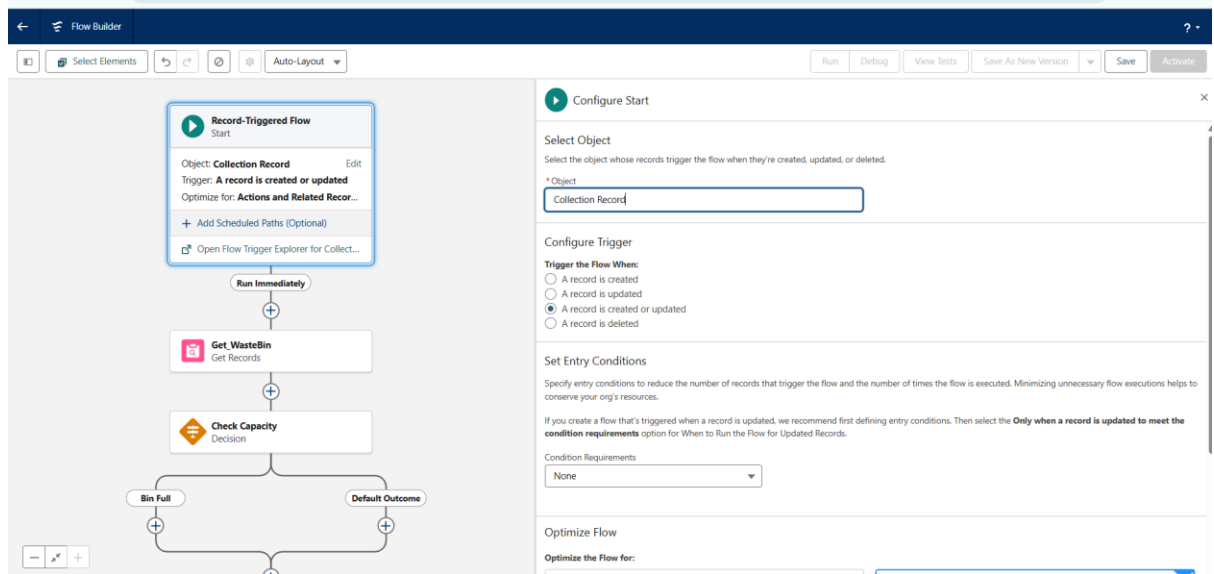
- Automate WasteBin status update when a CollectionRecord is created or updated.
- Reduce manual work and ensure accurate bin status tracking.

Expected Outcome: Collection of records automatically updates the corresponding WasteBin status.



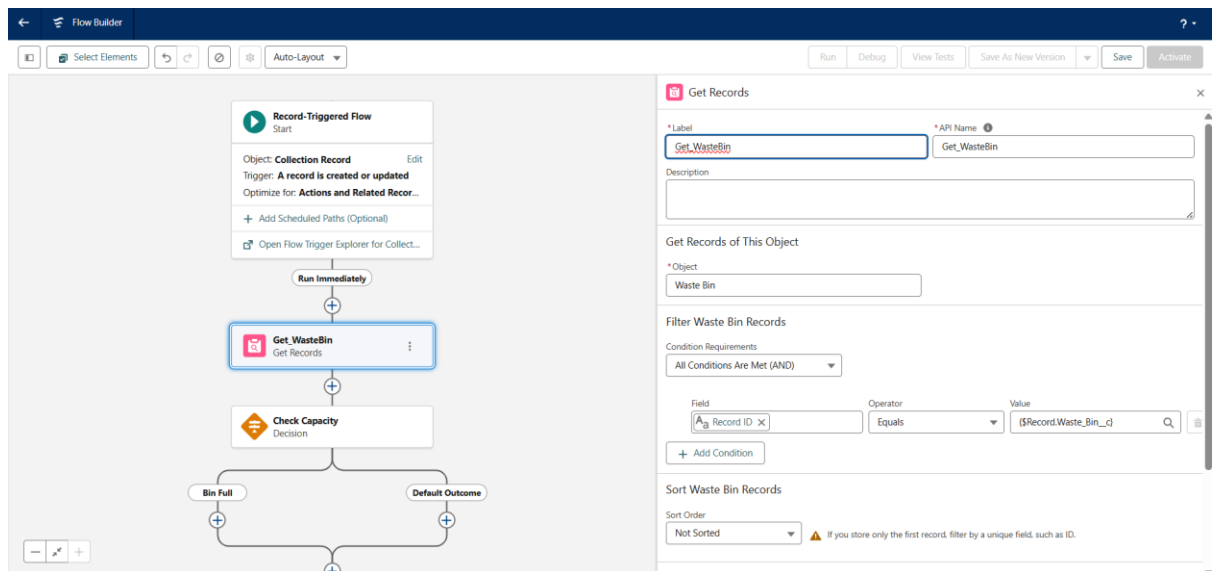
Step 2: Create Record-Triggered Flow

- Setup → Process Automation → Flows → New Flow → Record-Triggered Flow.
- Select **Object:** CollectionRecord__c
- Trigger the flow: **A record is created or updated**
- Entry Conditions: **None** (or as per requirement)
- Optimize Flow for **Fast Field Updates**



Step 3: Get WasteBin Record

- Add **Get Records** element → Label: Get_WasteBin
- Object: WasteBin__c
- Condition: Id = {!\$Record.WasteBin__c} (lookup from triggering CollectionRecord)
- Store all fields → Only the first record



Step 4: Check Bin Capacity (Decision Element)

- Add **Decision** → Label: Check Capacity
- Outcome 1: Bin Full → Condition: {\$Record.Quantity_Collected__c} >= Capacity__c

- Outcome 2: Default Outcome → Any other case

The screenshot shows the Flow Builder interface. On the left, a flow diagram is visible with the following steps: **Record-Triggered Flow** (Start) → **Run Immediately** → **Get WasteBin** (Get Records) → **Check Capacity** (Decision). The decision node has two paths: **Bin Full** and **Default Outcome**. On the right, the configuration for the **Check Capacity** decision is shown. It includes a label 'Check Capacity', an API name 'Check_Capacity', and a description field. Under 'OUTCOMES', the 'Bin Full' outcome is configured with the condition 'All Conditions Are Met (AND)' and a resource path: `..c > Quantity Collected (Litres)` compared to `..from Get_WasteBin > Capacity` using the 'Greater Than or Equal' operator. The 'When to Execute Outcome' is set to 'If the condition requirements are met'.

Step 5: Update WasteBin Status

- Add **Update Records** → Label: Update_WasteBin_Status
- Record to update: Waste Bin from Get_WasteBin
- Set Field Values:
 - Field: Status__c
 - Value: Full (or based on logic)

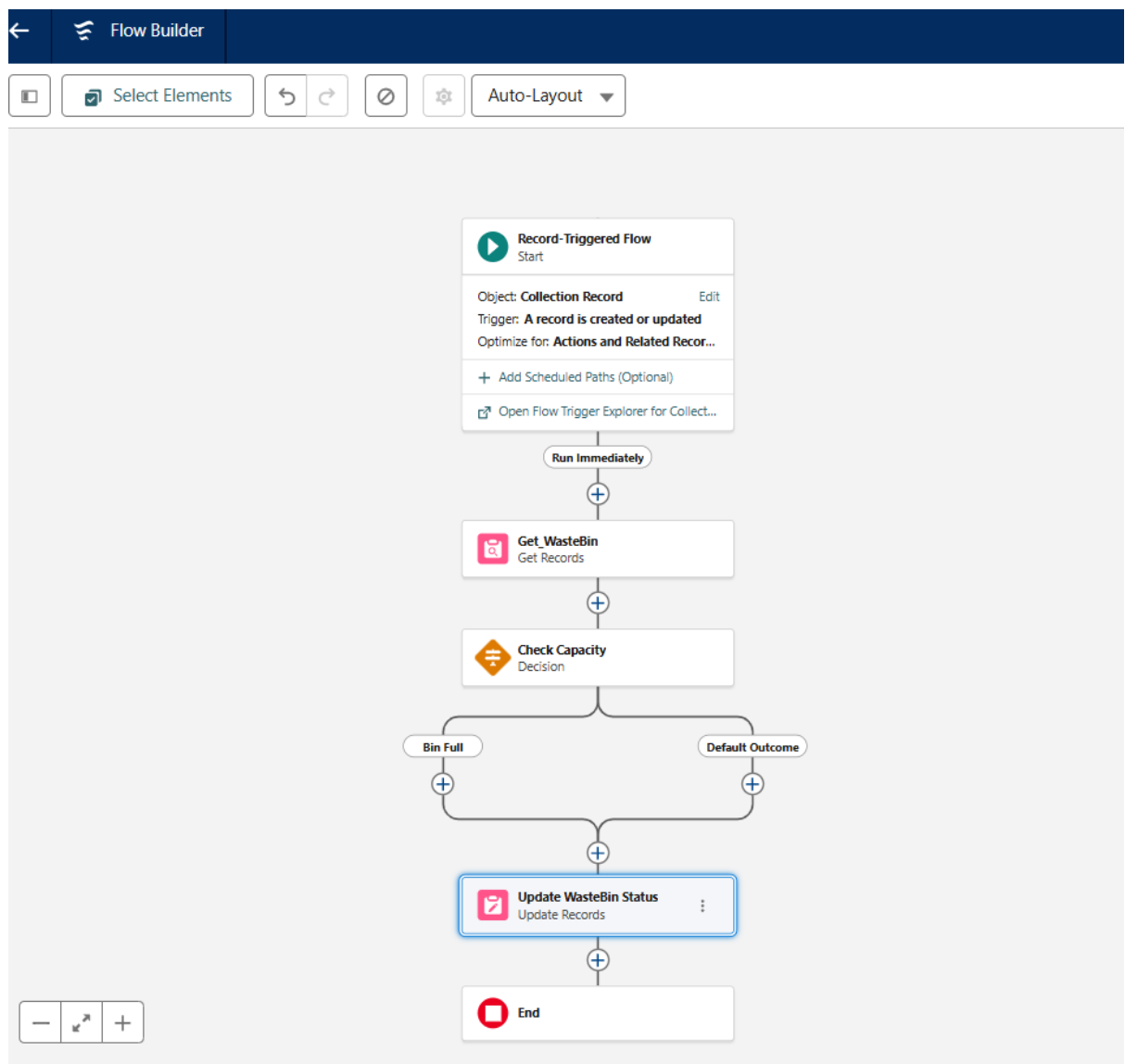
The screenshot shows the Flow Builder interface with the 'Update Records' node configuration. The flow diagram on the left now includes the **Update WasteBin Status** (Update Records) node at the end of the flow. The right-hand pane shows the configuration for this node. It includes a section 'How to Find Records to Update and Set Their Values' with radio buttons for different selection methods, where 'Specify conditions to identify records, and set fields individually' is selected. Under 'Update Records of This Object Type', the object is set to 'Waste Bin'. The 'Filter Waste Bin Records' section shows a condition: 'Record ID' equals '..from Get_WasteBin > Record ID'. The 'Set Field Values for the Waste Bin Records' section shows the field 'Status__c' being set to the value 'Full'.

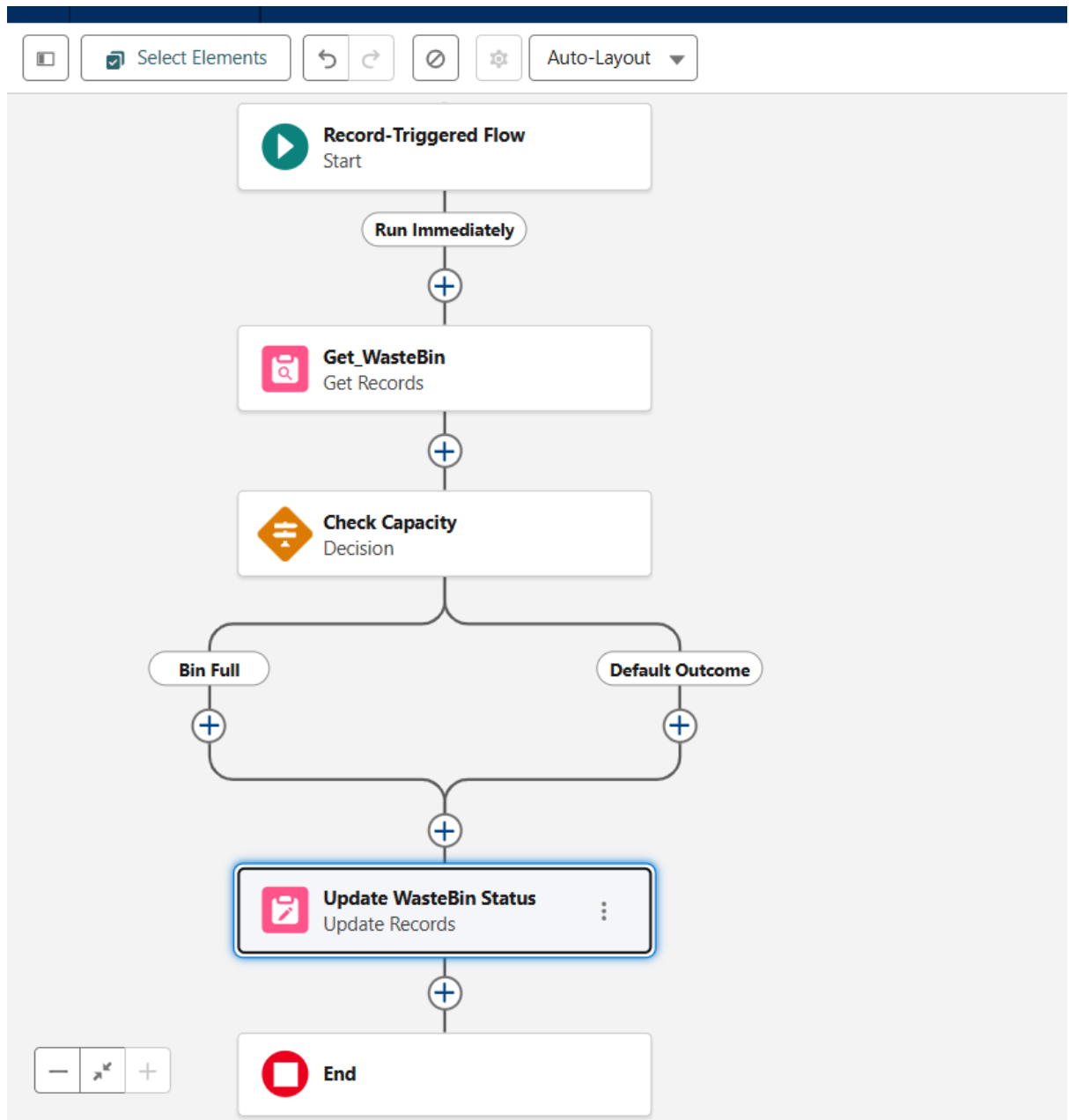
Step 6: Connect Flow Elements

- Start → Get_WasteBin → Check Capacity → Update_WasteBin_Status → End
- Connect both decision outcomes if needed

Step 7: Save & Activate Flow

- Flow Label: Update Waste Bin Status
- API Name: Update_Waste_Bin_Status
- Description: "Automatically updates WasteBin__c status to Full when CollectionRecord__c is created/updated"
- Save → Activate





Expected Outcome (Phase 4 Completion)

- WasteBin status automatically updates based on CollectionRecord quantity.
- Reduces manual errors and improves operational efficiency.
- Flow ready for testing in Sandbox/Dev Org.