Claude Haiku model on Amazon Bedrock, and completed the UI configuration as explained in the prerequisites section before proceeding.

To build it quick, we have placed 2 documents in the S3 bucket with the name containing `s3useruploadbucket`.

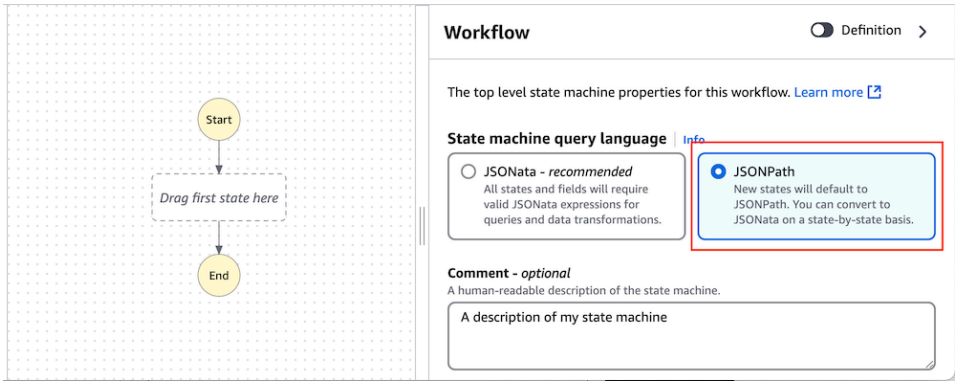
In this workflow

- You will iterate through the documents (objects) in S3 using Distributed Map.
- You will then use a parallel state in Step Functions to perform extraction and summarization in parallel by invoking the respective Lambda function. The reason for using a parallel state is to simultaneously execute the extraction and summarization as they are independent of each other.

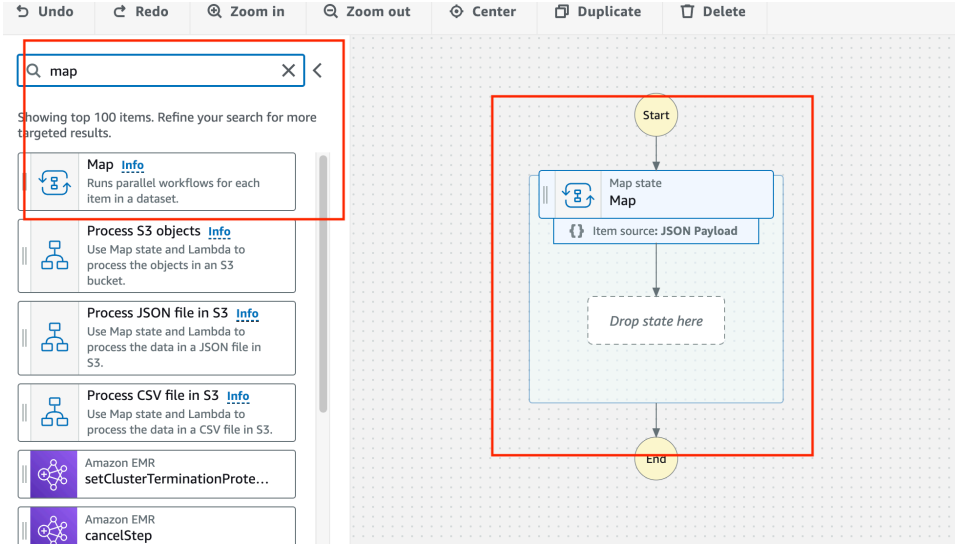
> **Why distributed map?** There can be 1000s of documents that needs to be processed. Distributed map is a purpose built iterator that can iterate millions of items in a collection at an unparallel concurrency, run business logic on these items and support configurations such as batching, concurrency and failure tolerance. If you want to learn more about distributed map, try this workshop later.

Using visual builder to build the workflow

1. Navigate to AWS Step Functions in your AWS Console. Make sure you are in the correct region.
2. If you are not on the `State Machines` page, choose `State machines` on the left side hamburger menu icon and then select **Create state machine**
3. On the *Choose a template* overlay, choose the **Blank** template and select **Select**.
4. Choose **JSONPath** as query language on the right side of the workflow configuration.



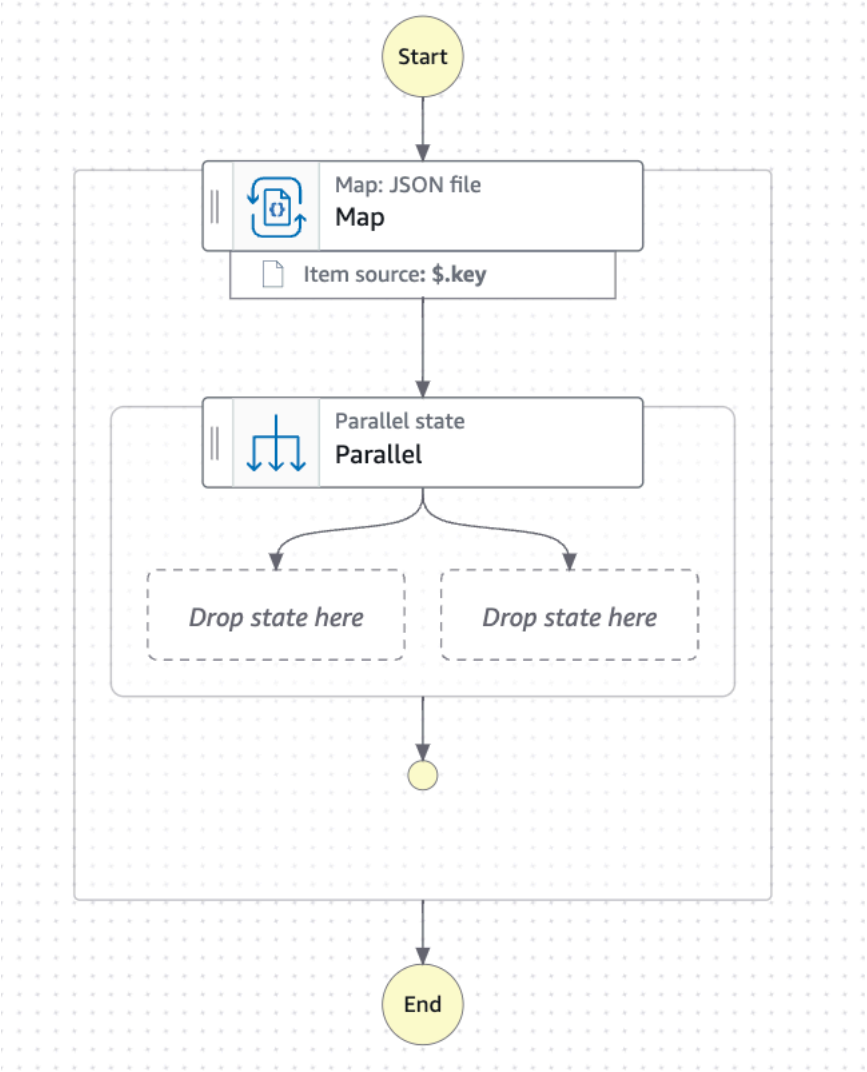
5. Under the Action tab, search for **Map** and drag and drop the map action into the Workflow Studio Canvas.



6. Click on the Map state and select **Distributed - New** as the processing mode on the right pane and configure the Distributed Map state with the following values.

Setting	Value	Comment
State name	`Document Processing`	
Processing mode	Distributed - new	
Item source	Amazon S3	
S3 item source	JSON File in S3	
S3 bucket	Choose `Get bucket and prefix at runtime from state input`	
Bucket Name	`$.input_bucket`	
Key	`$.key`	
Set concurrency limit	1	Because of workshop restrictions, you will run the process in serial though Distributed map can run at a concurrency of 10K!
Child execution type	Standard	If an iteration can complete in 5 minutes, prefer `Express` workflow. For the workshop, you choose `Standard` for the console experience.

7. Under the Action tab, search for **Parallel** Action. Drag and drop the parallel action into the Workflow Studio Canvas. Parallel step runs the actions in the step in parallel.



Ensure that the Parallel state is nested within the Map state to properly execute parallel actions for each item in the map iteration.

▶ **Incorrect Placement**

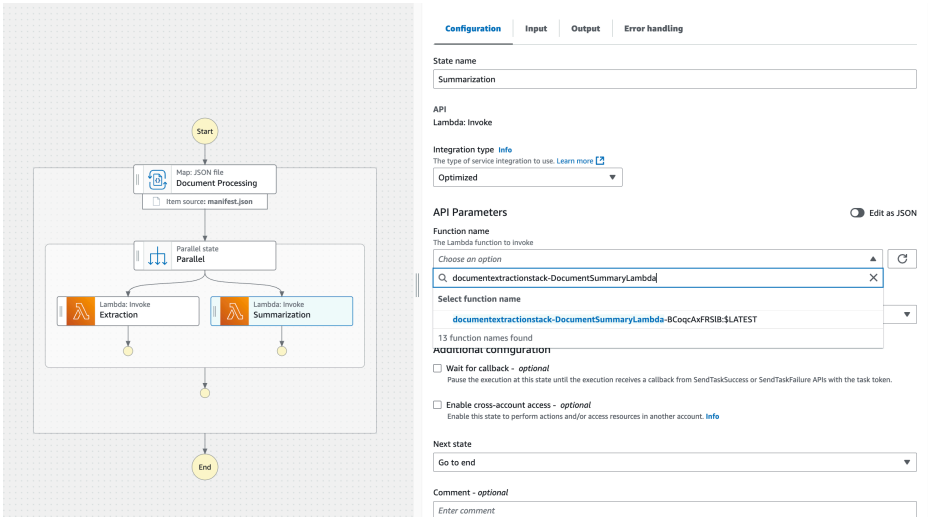
8. Under the Action tab, search for **Lambda Invoke**. Drag and drop the **Lambda Invoke** action in one of the parallel states. Select that Lambda state within the Distributed Map state. Configure the state with the following values.

Setting	Value
State name	`Extraction`
Function name	Search for `DocumentExtractionLambda` and select the function

9. Repeat the previous step and drop another Lambda invoke into the parallel state. Configure the state with the following values.

Setting	Value
State name	`Summarization`
Function name	Search for `DocumentSummaryLambda` and select the function

10. Your workflow should look similar to the screenshot below.



Configure the workflow

1. Select the Config tab next to the state machine name at the top of the page and edit the state machine name: `DocumentProcessingWorkflow`
2. Choose an existing role containing `DocumentProcessingSfnRole` as `Execution role`
3. Leave the rest of the defaults and select **Create**.

> In the next section, you'll execute the workflow and view the results of the document extraction and summarization process.

Previous Next