Lesson 02 Demo 03

Managing Memory Usage in MongoDB

Objective: To manage the memory usage of the databases by analyzing and optimizing queries to improve database performance and enhance the customer experience

Tools Required: MongoDB Compass

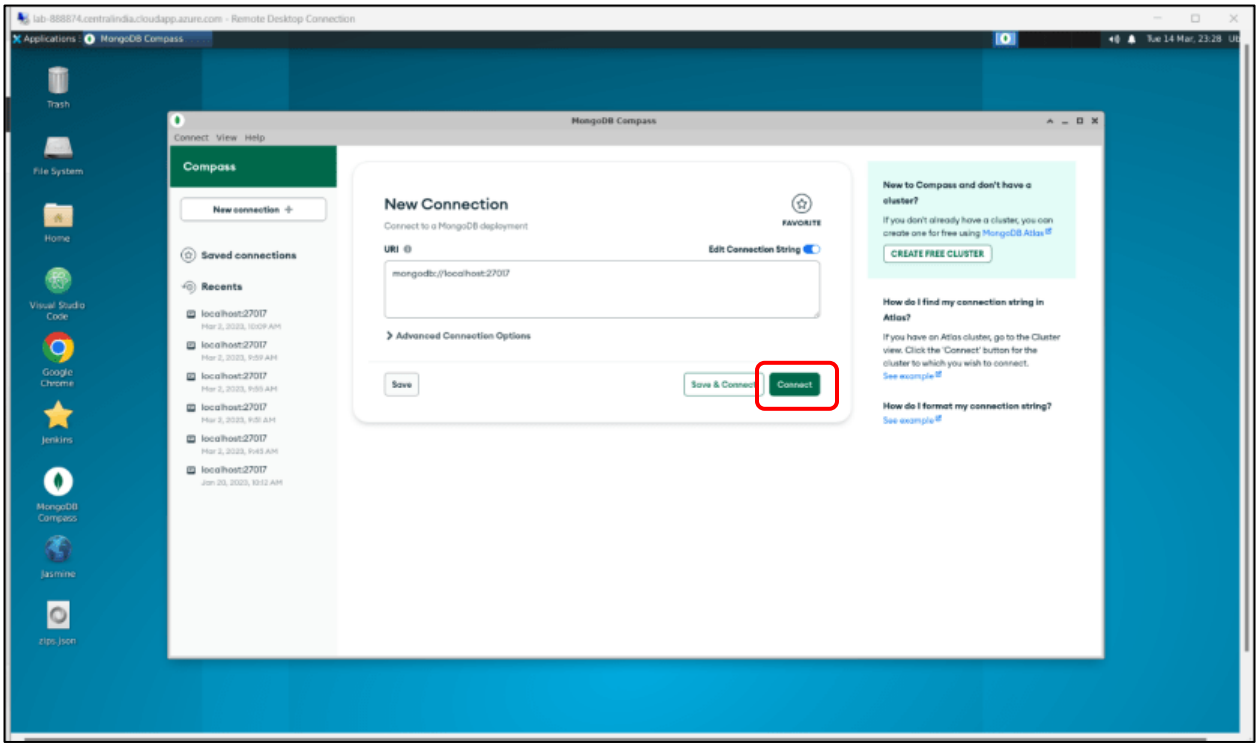
Prerequisites: Knowledge of MongoDB

Steps to be followed:

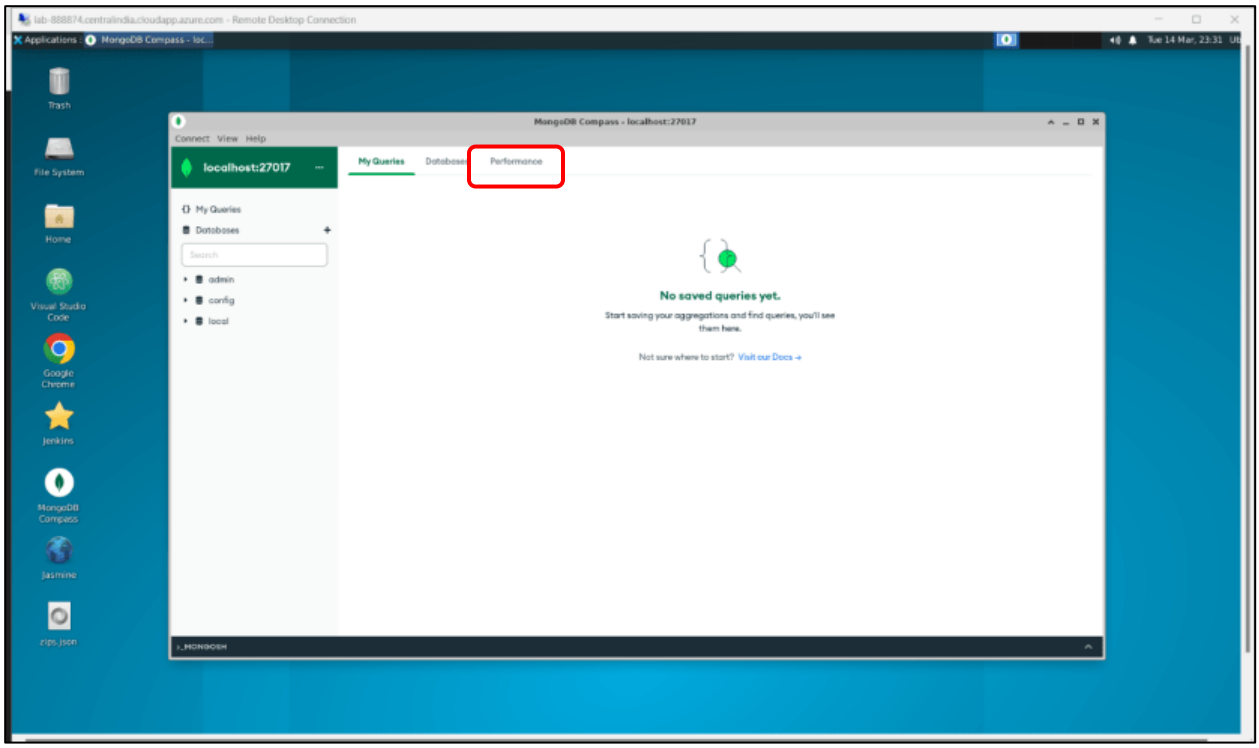
1. Check the memory usage by analyzing and optimizing queries
2. Use the aggregation pipeline to perform aggregation operations

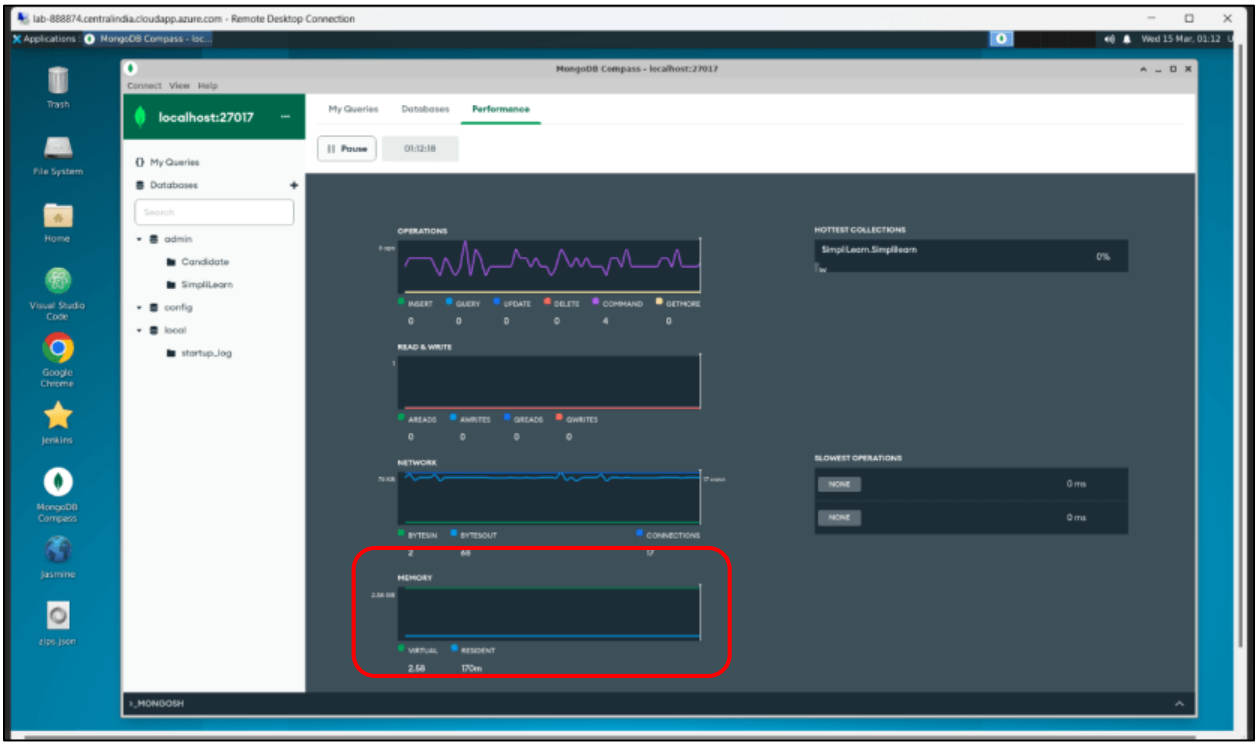
Step 1: Check the memory usage by analyzing and optimizing queries

1.1 Connect to the **MongoDB** server

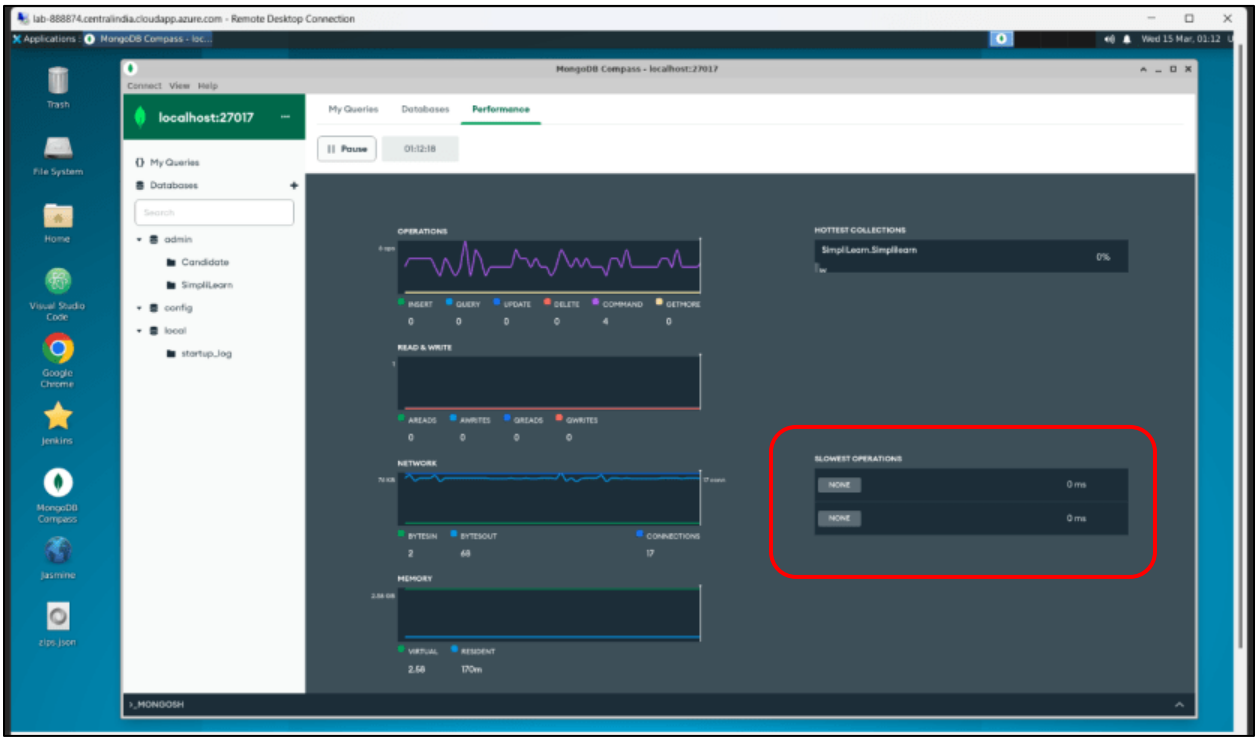


1.2 Click on the **Performance** tab from the top of the **MongoDB** window





At the bottom, you can see the memory usage of your database.

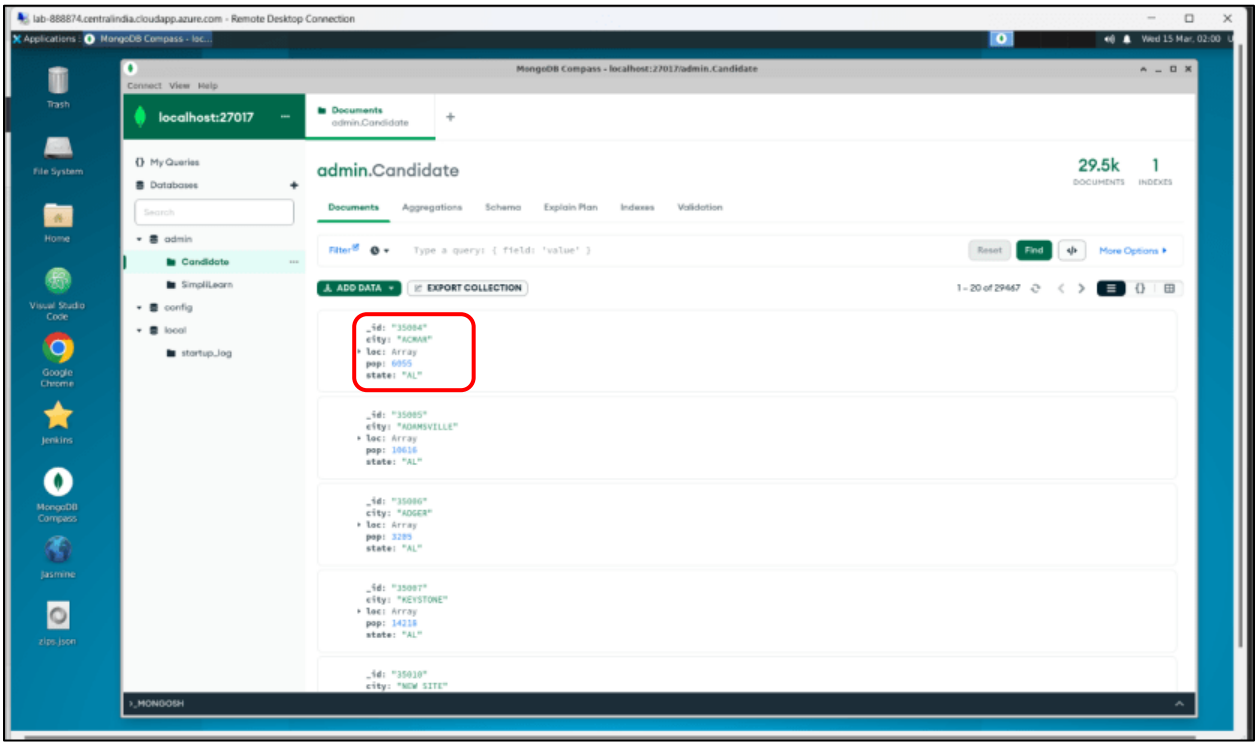


On the right-hand side of the window, you can see the list of the slowest operations. To optimize these operations, you can use techniques such as indexing, aggregation pipelines, or modifying the query parameters.

Here, all the operations are running fine.

Step 2: Use the aggregation pipeline to perform aggregation operations

2.1 Select the database collection on which you want to perform the aggregation



As you can see, there is sample data in the MongoDB compass created. For example:

_id: "35004" //This field holds the zip code as a string.

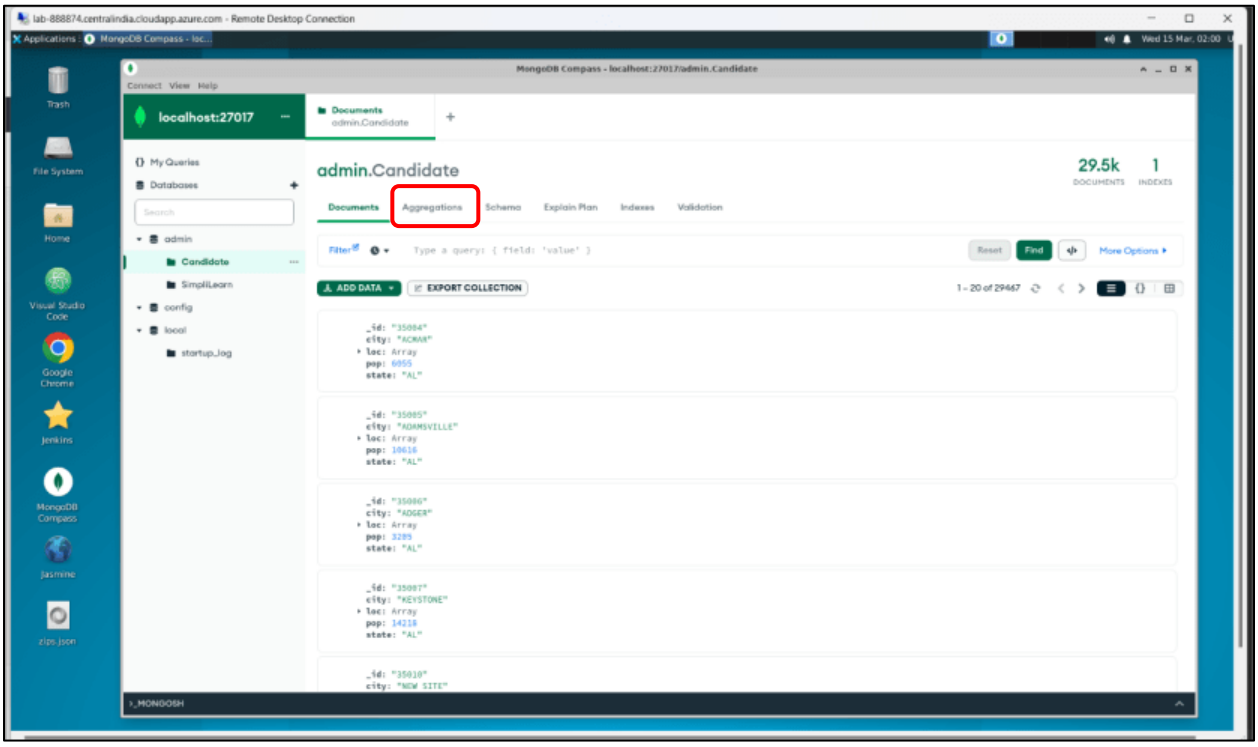
city: "ACMR" //This field holds the city name.

loc: Array // This field holds the location as a longitude-latitude pair.

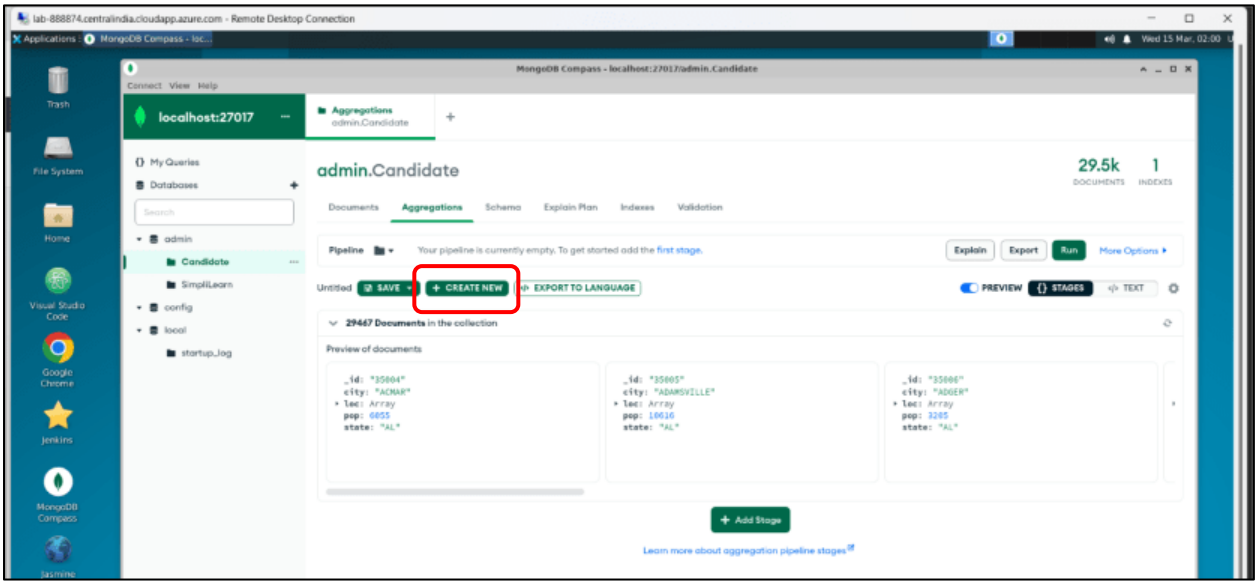
pop: 6055 // This field holds the population.

state: "AL" //This field holds the two-letter state abbreviation.

2.2 Click on the **Aggregations** tab

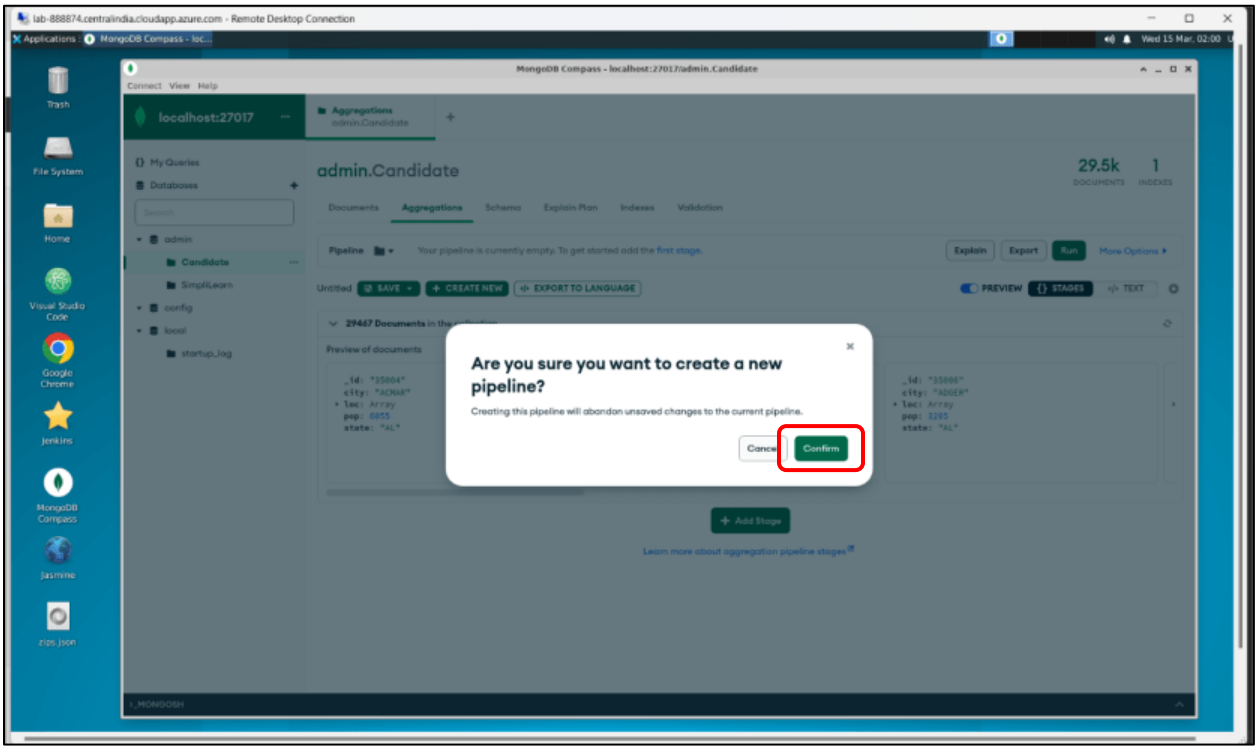


2.3 Click on **+ CREATE NEW**

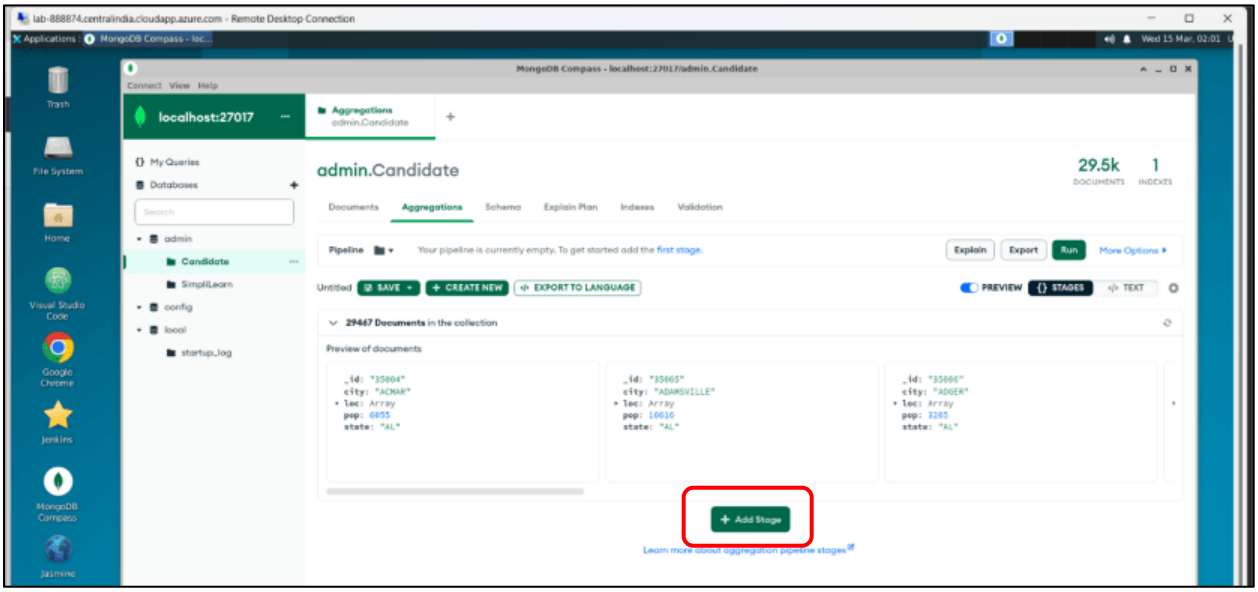


A pop-up window will appear to warn that the unsaved changes will be lost during the creation of a new pipeline.

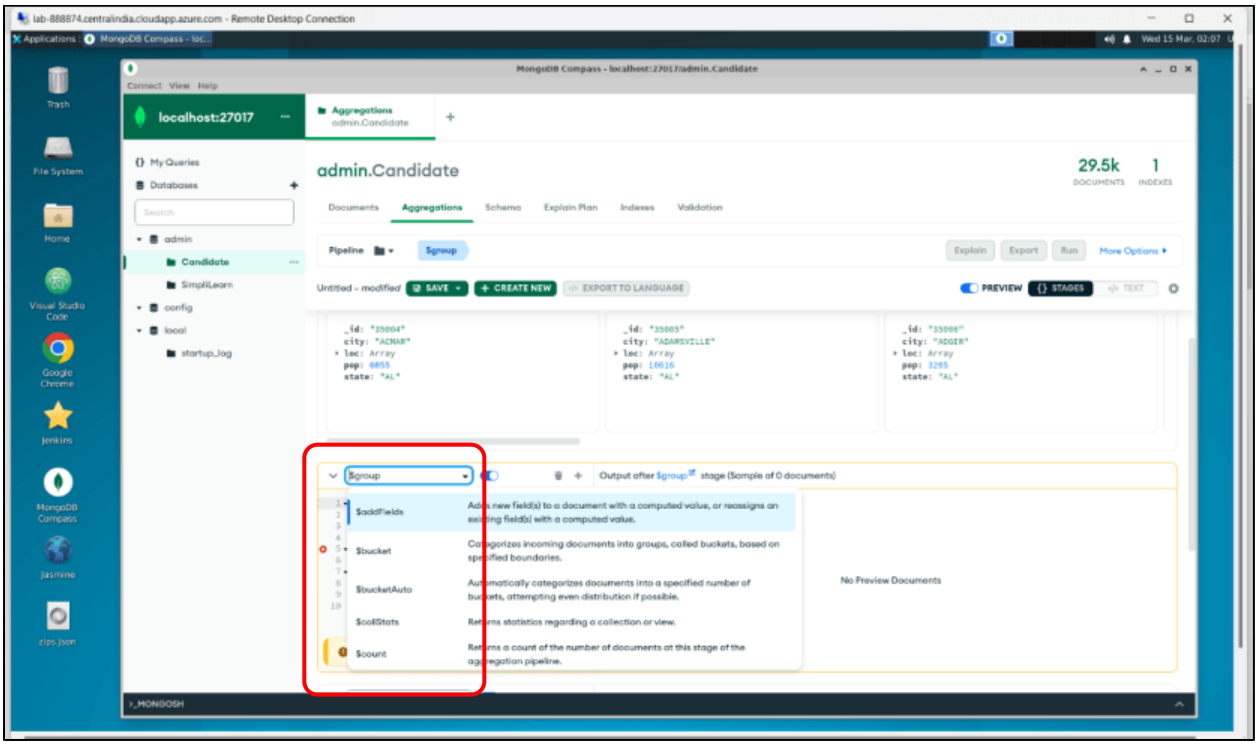
2.4 Click **Confirm** to proceed



2.5 Click **+ Add Stage** to build an aggregation pipeline

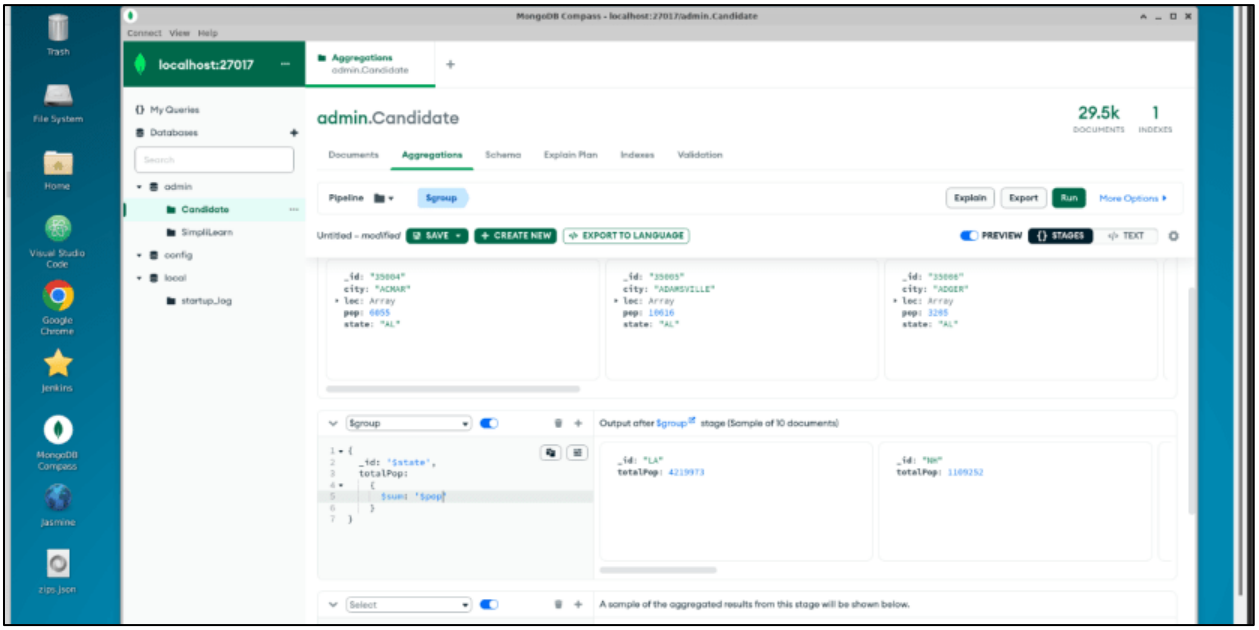


2.6 Select the stage from the drop-down list and filter by **$group** to group documents based on a specific field



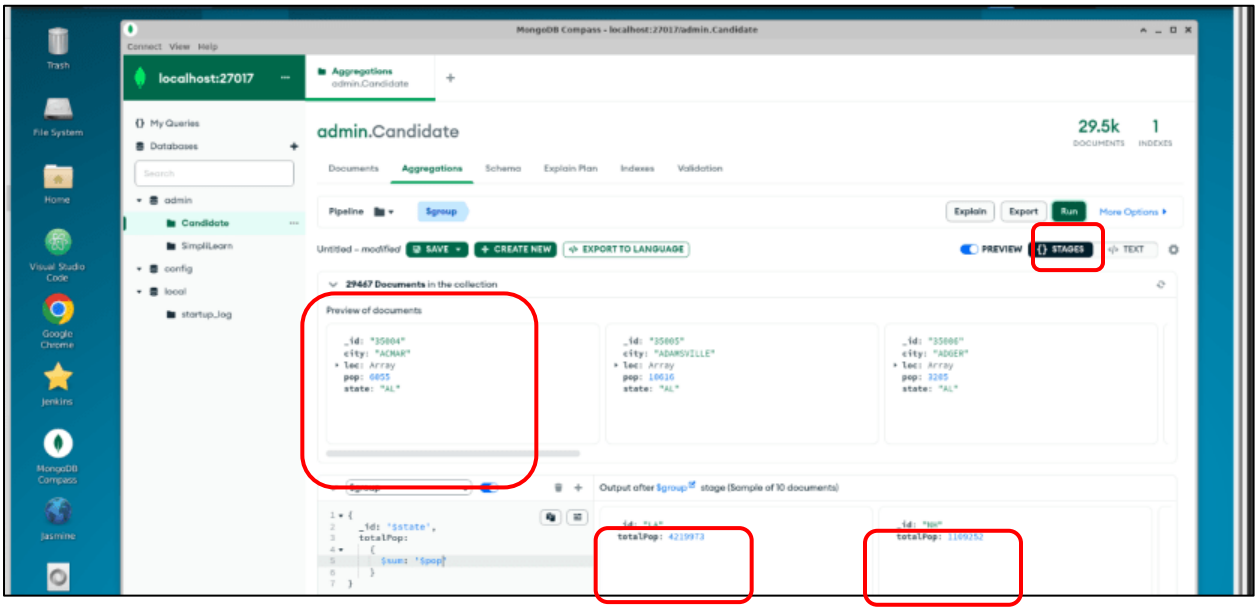
2.7 Write the following commands to specify the field where the results will be grouped:

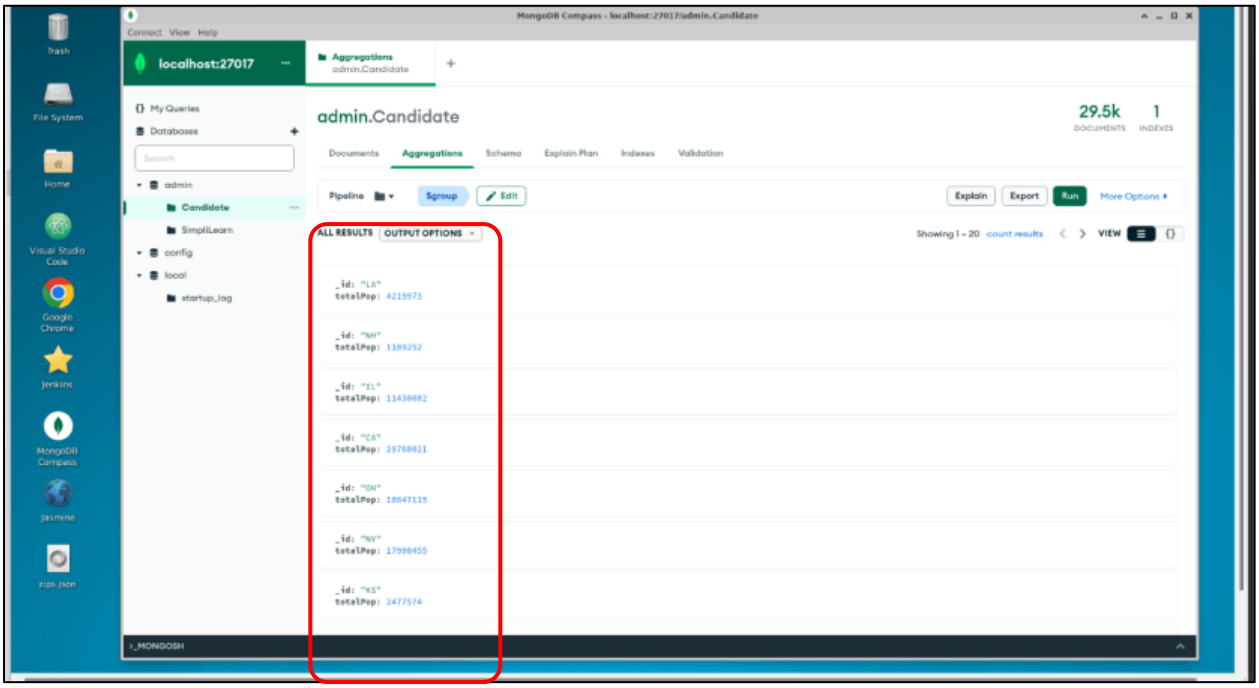
```
{
  _id: “$state”,
  totalPop:
     {
       $sum: ‘$pop’
     }
}
```



Here, the user can group the results based on the states and calculate the total population of all the states individually.

2.8 Click **Run** to view the results of the pipeline





By following these steps, you have successfully managed the memory usage of the databases by analyzing and optimizing queries to improve database performance and enhance the customer experience.