**Project Report**

**Project Title**

**Customer Intent Prediction & Pattern Discovery for e-Commerce businesses**

**Team Leader:**

Shivanoor Vignesh 21R11A0597

**Team Members:**

G.Sandhya Rani 21R11A0569

Dhathri 21R11A0545

Hasini 21R11A0518

Pooja 21R11A0501

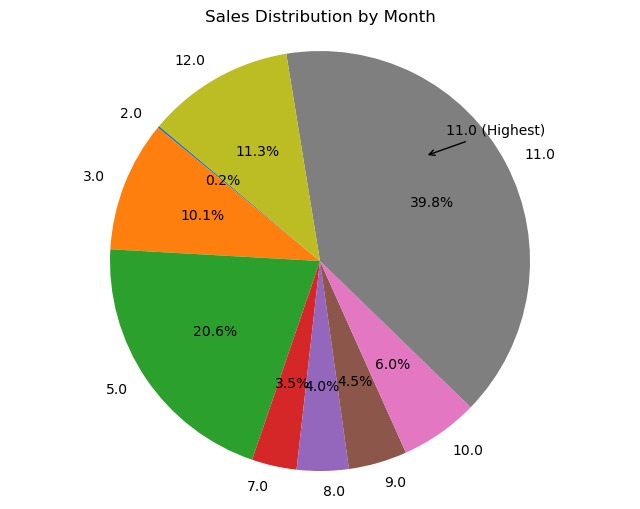
PSBN Sriya 21R11A0595

N Sai Charan 21R11A0588

MD.Akhbar Ali 21R11A0586

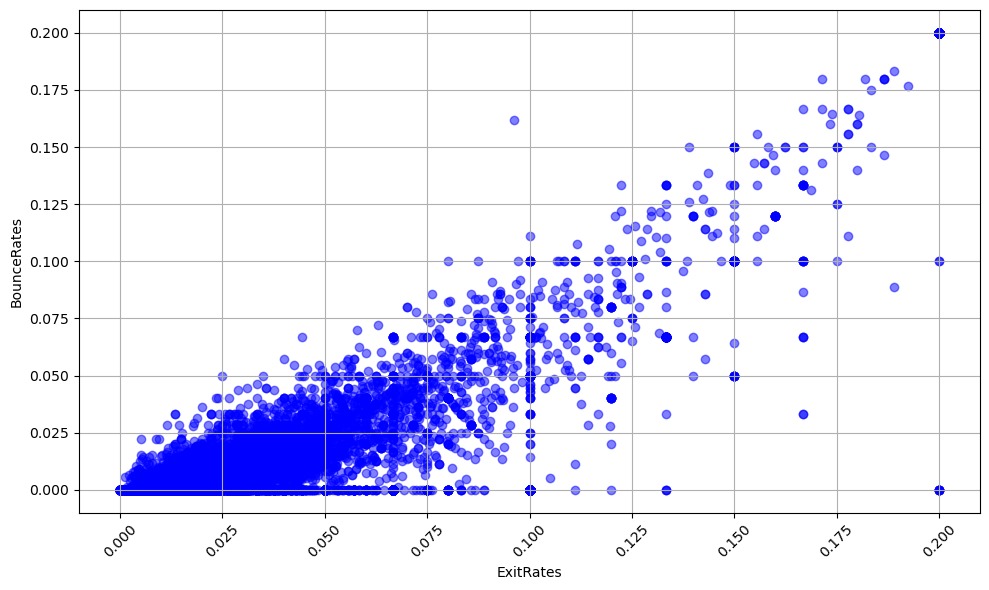
**EXPLORATORY DATA ANALYSIS**

**Sales Distribution By Month**



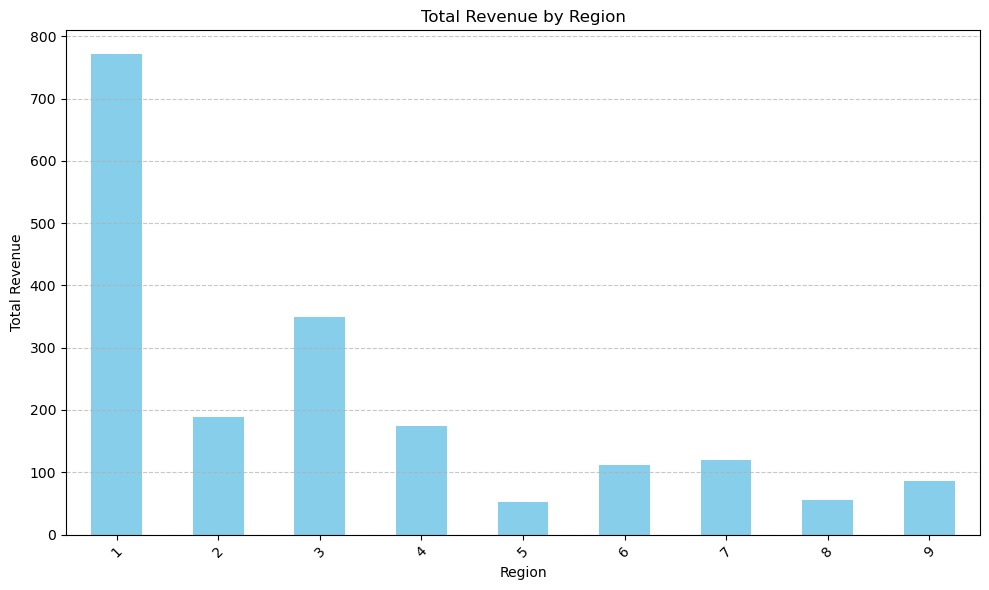
Month of November has the highest sales distribution while febuary has lowest sales

**Bounce Rates vs exit rates**



The above scatter plot has more density between a ratio of 0.050 (on the y-axis) and 0.075 (on the x-axis) in terms of the bounce rates and exit rates.

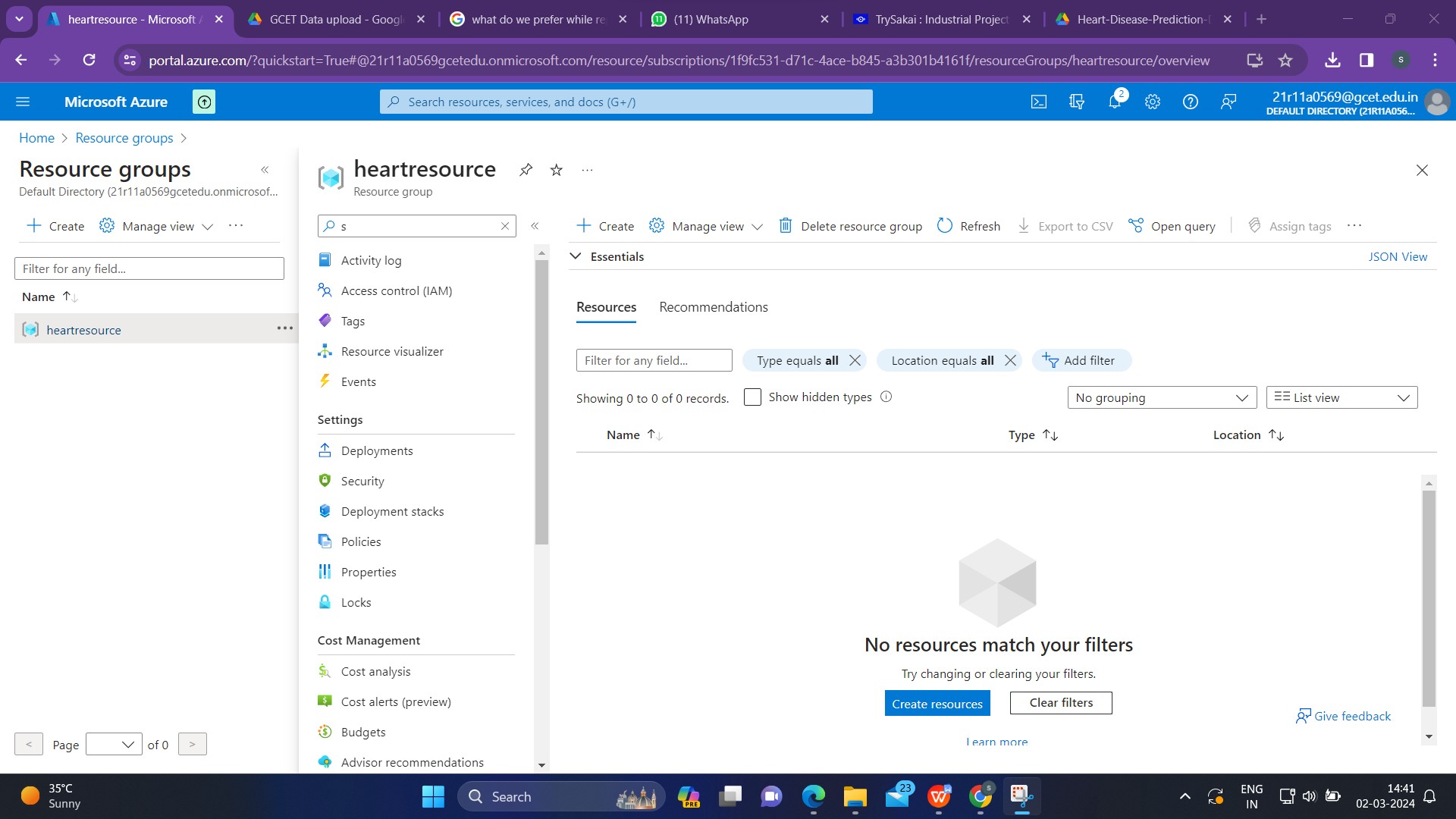
**Total revenues by Region**



The above histplot says that Region 1 has the highest total revenue greater than 750

**Modelling**

**Resource group creation:**



To create a resource group in a cloud platform like Microsoft Azure, one has to generally follow these steps:

1.Sign in to the Cloud Platform: Go to the Azure portal or the relevant cloud provider's portal.

2.Search for Resource Group:In the Azure portal, we'll typically find a search bar at the top. Enter "Resource Group" to locate the service.

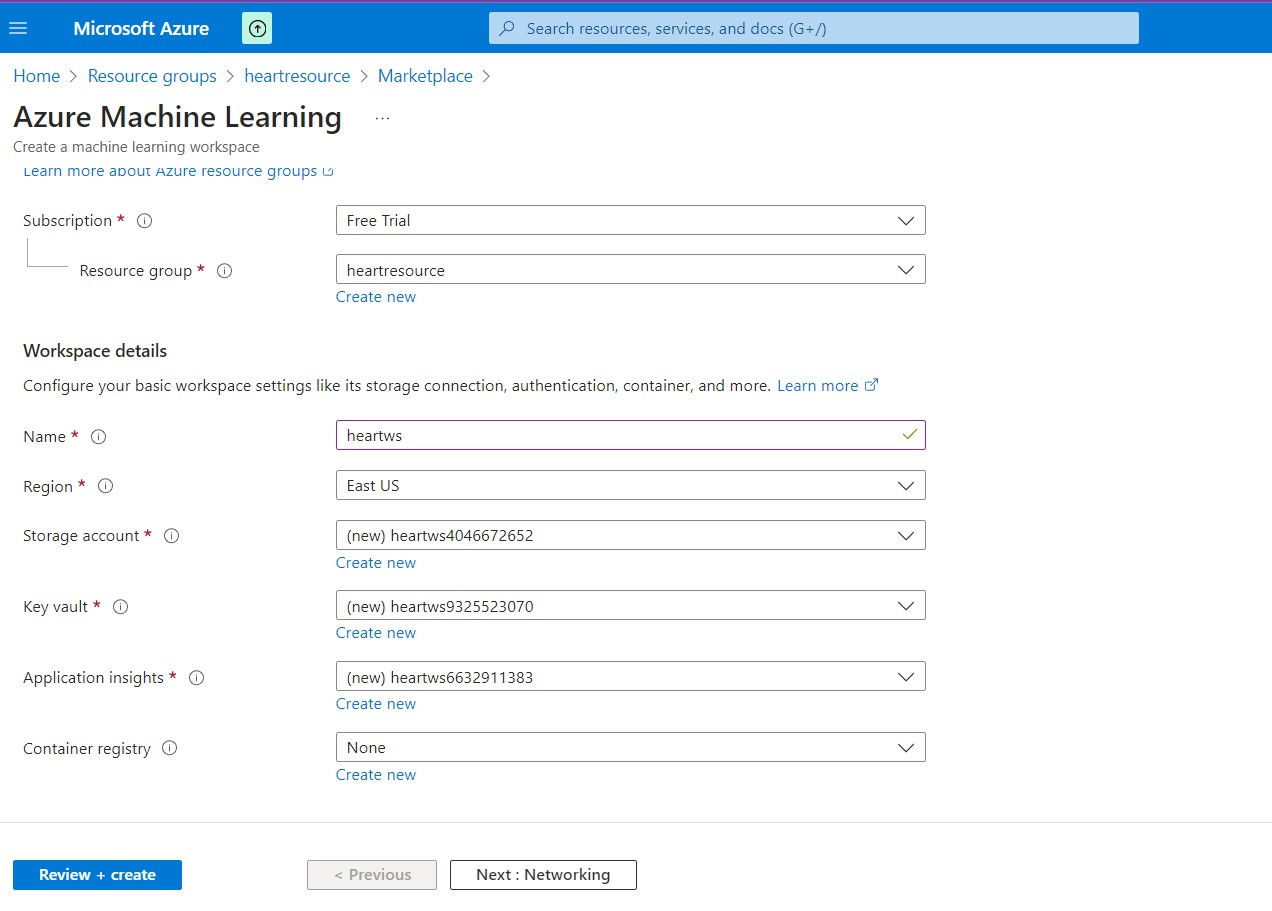
3.Click on Create:Once we find the Resource Group service, click on it.

4.Providing Information:we'll usually see a "Create" button. Click on it.

5.Enter a name for your resource group.Choose the location (in this case, East US).

6.Complete the Creation.

**Creating a Workspace:**



1.After creating resource group, go back to the Azure portal.

2.Look for the search bar at the top, and enter "Azure Machine Learning" to find the service.

3.Click on the Azure Machine Learning service in the search results.

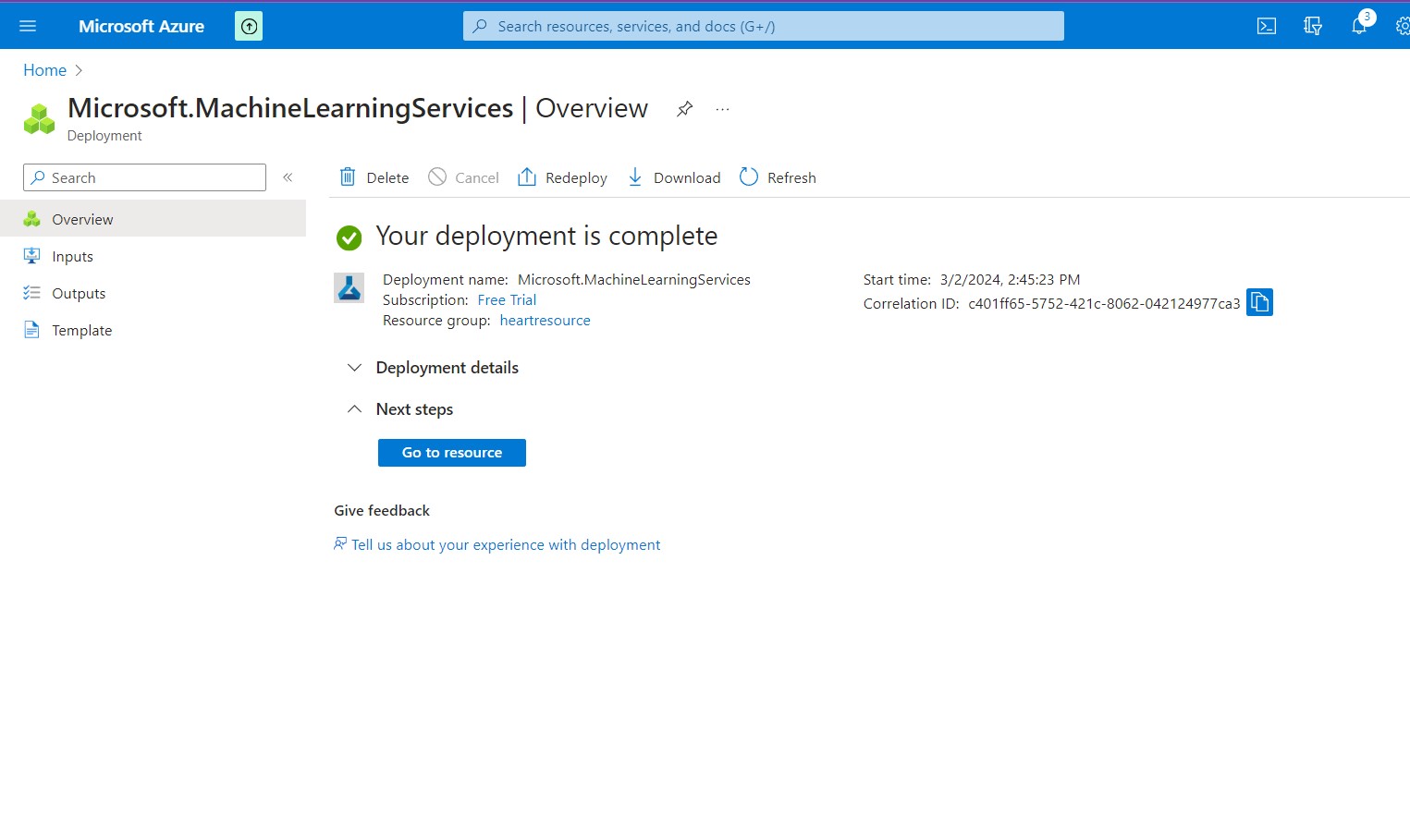
4.We'll typically find a "Create" or "Add" button. Click on it to start the process of creating an Azure Machine Learning workspace.

5.Provide Workspace Details:

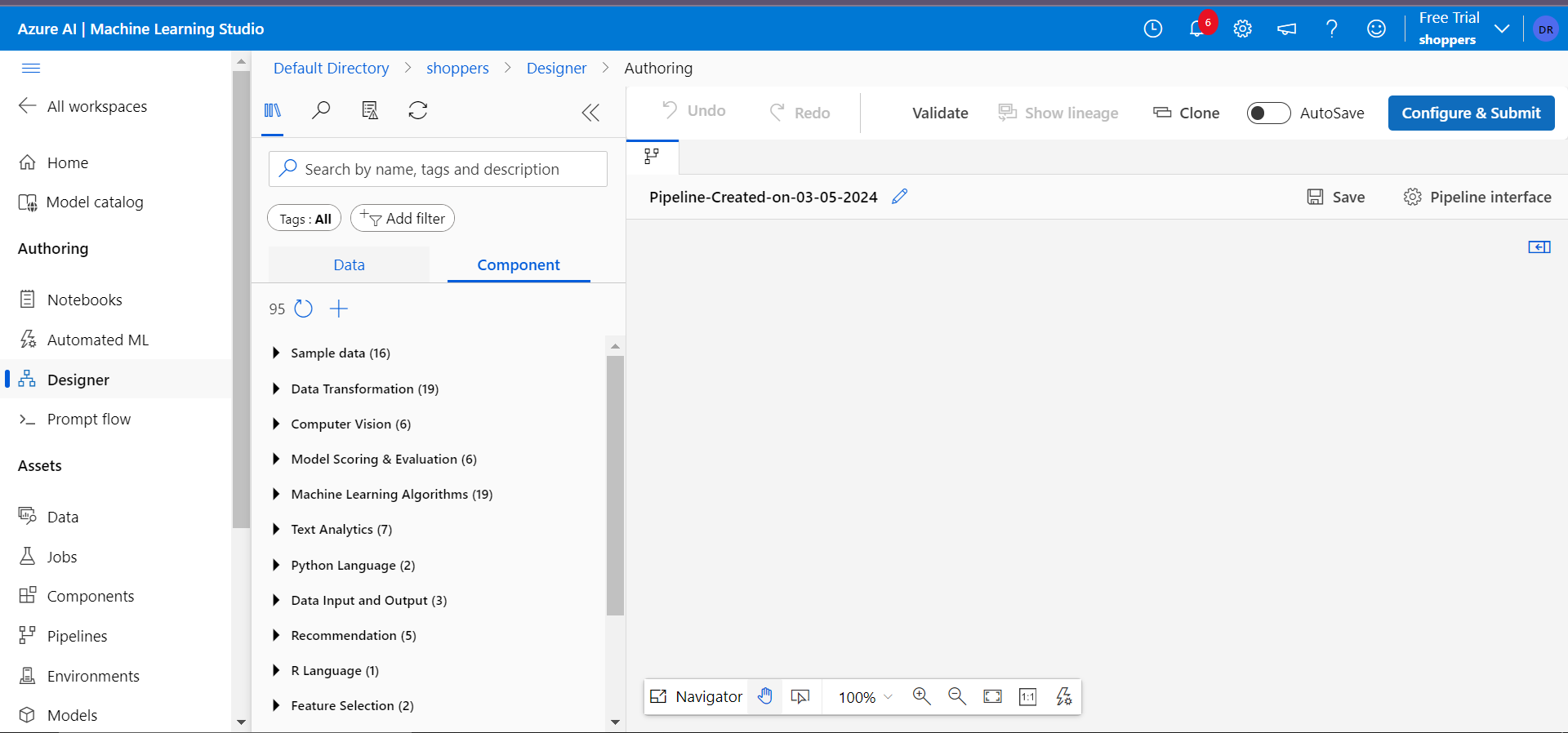
* During the creation process, you'll be prompted to provide details for your Azure Machine Learning workspace.
* Enter a unique name for your workspace.
* Select the subscription you want to use.
* Choose the resource group you created earlier.

6.Specify the region or location (East US).

7.Complete the Creation



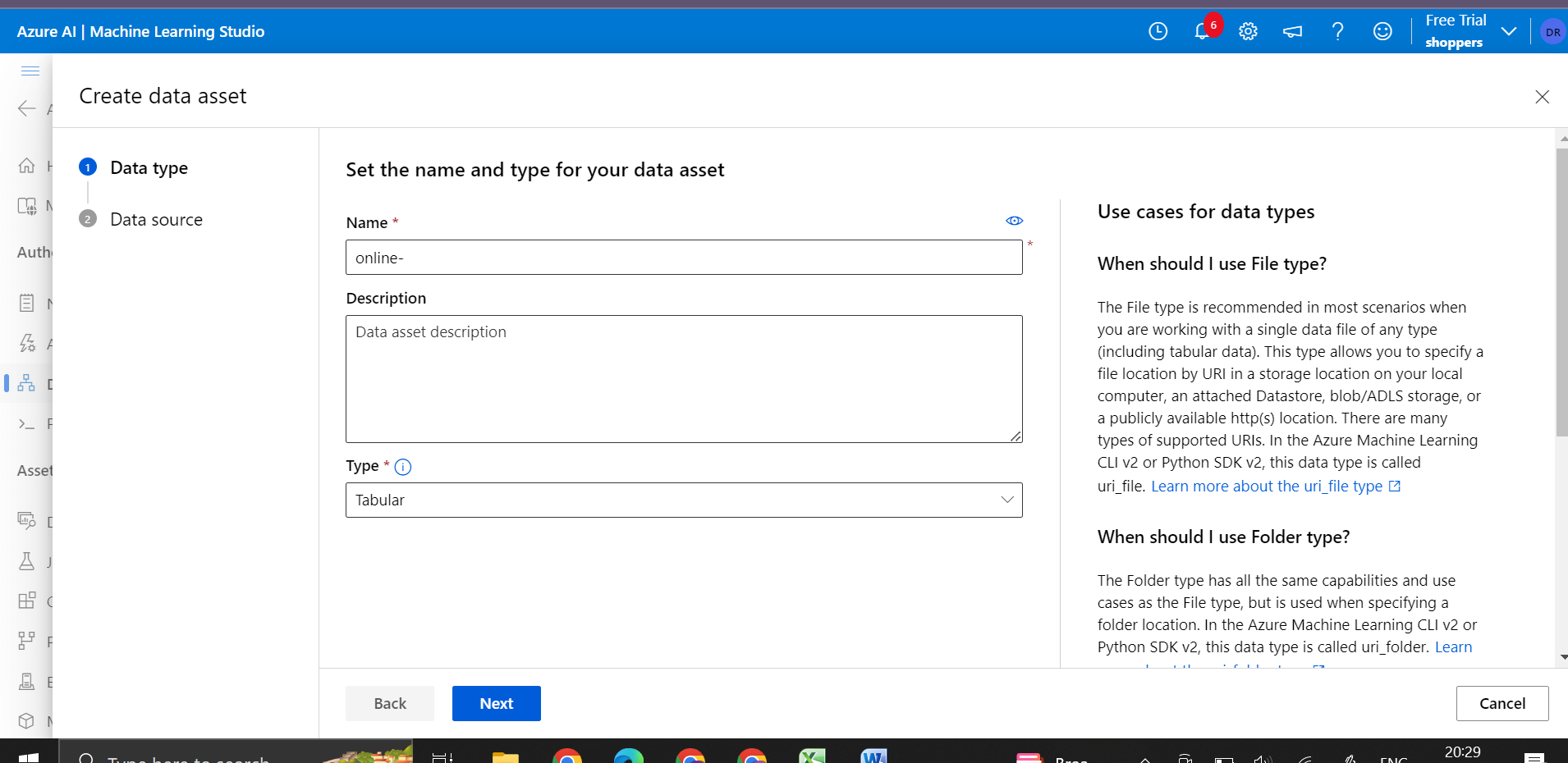
**Data Uploading**



After successfully completing the deployment,go to the workspace which we have created and launch it.

Within the Azure Machine Learning studio, look for a section or tab labeled "Design" or "Designer." Click on it.

**Data type:**

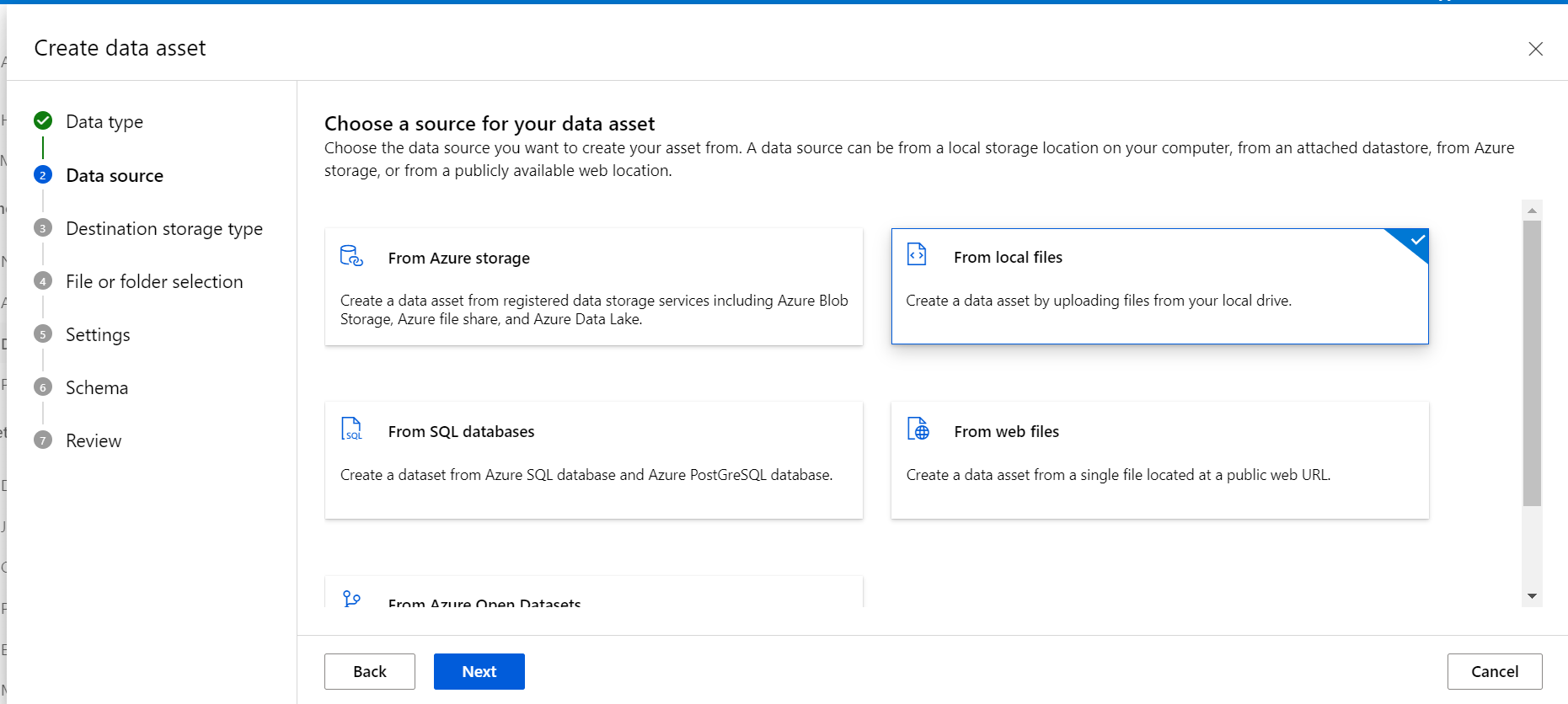


Within the Designers section, locate the option for working with data.

Once we're in the data section,create a new data asset by clicking on “Create” button.

Provide the necessary details for data asset, including the name, description(optional).

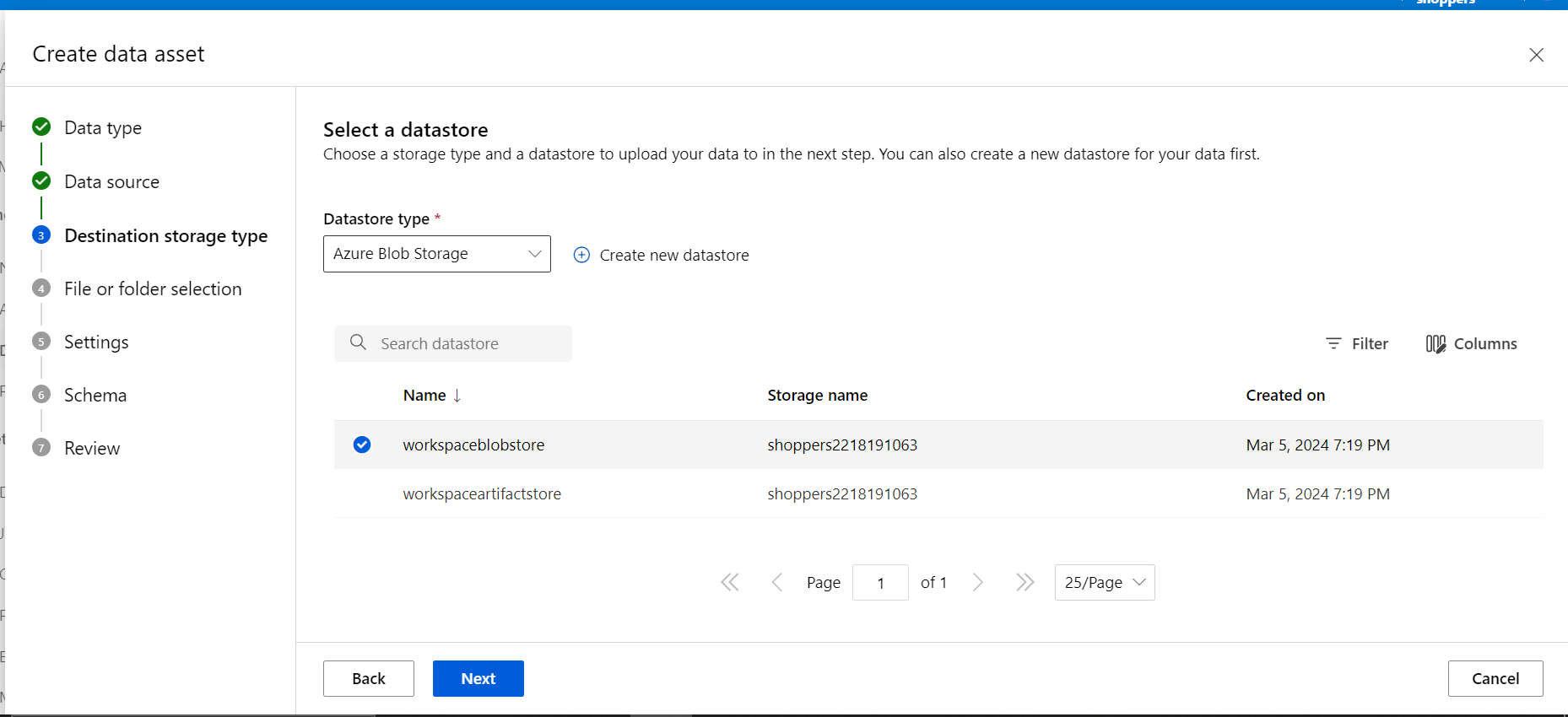
**Data source:**



For choosing a data source,select the option “From local files”.

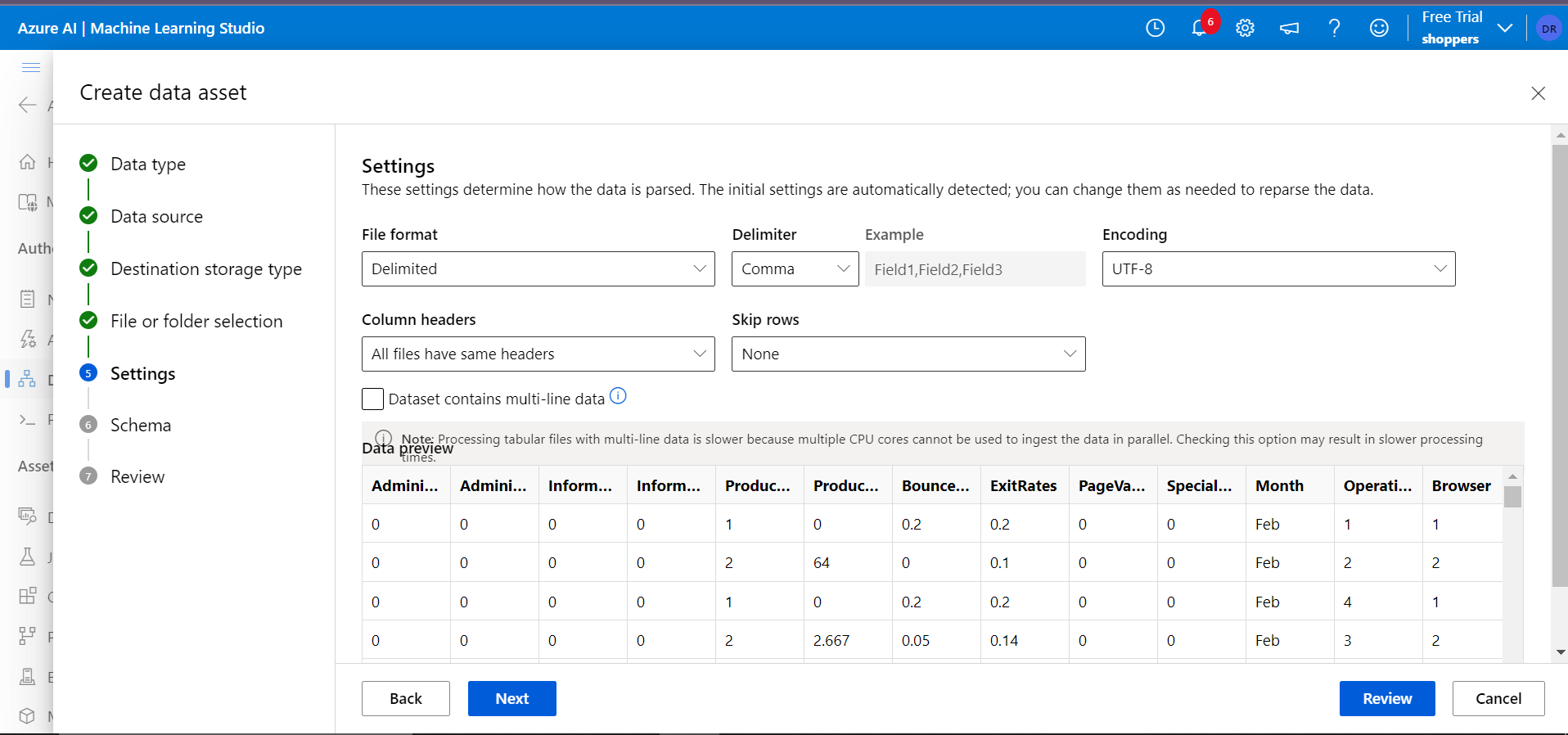
we'll likely see a button that enables us to upload files. Click on it to browse our local computer and select the CSV file we want to upload.

Destination Storage type:



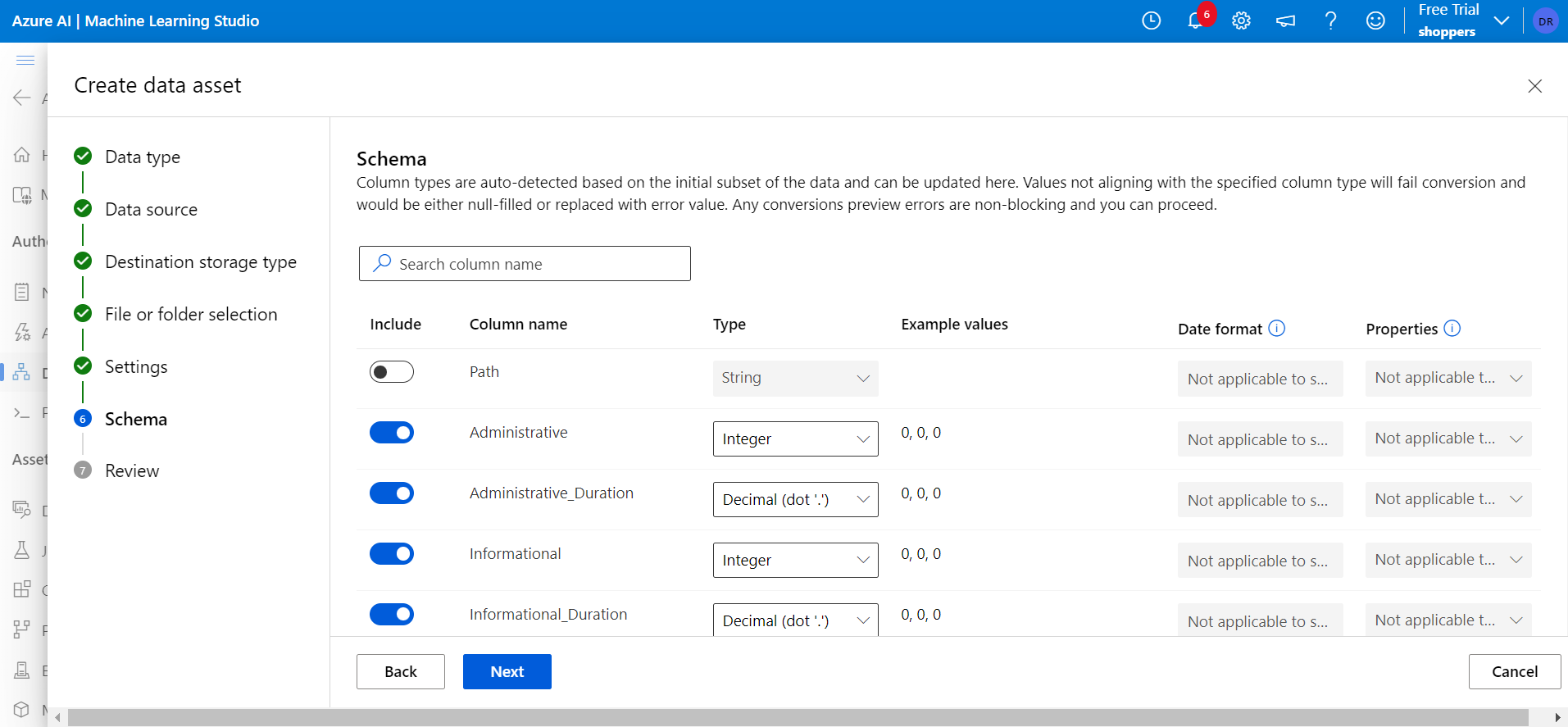
Always select Azure blob store.

Settings:

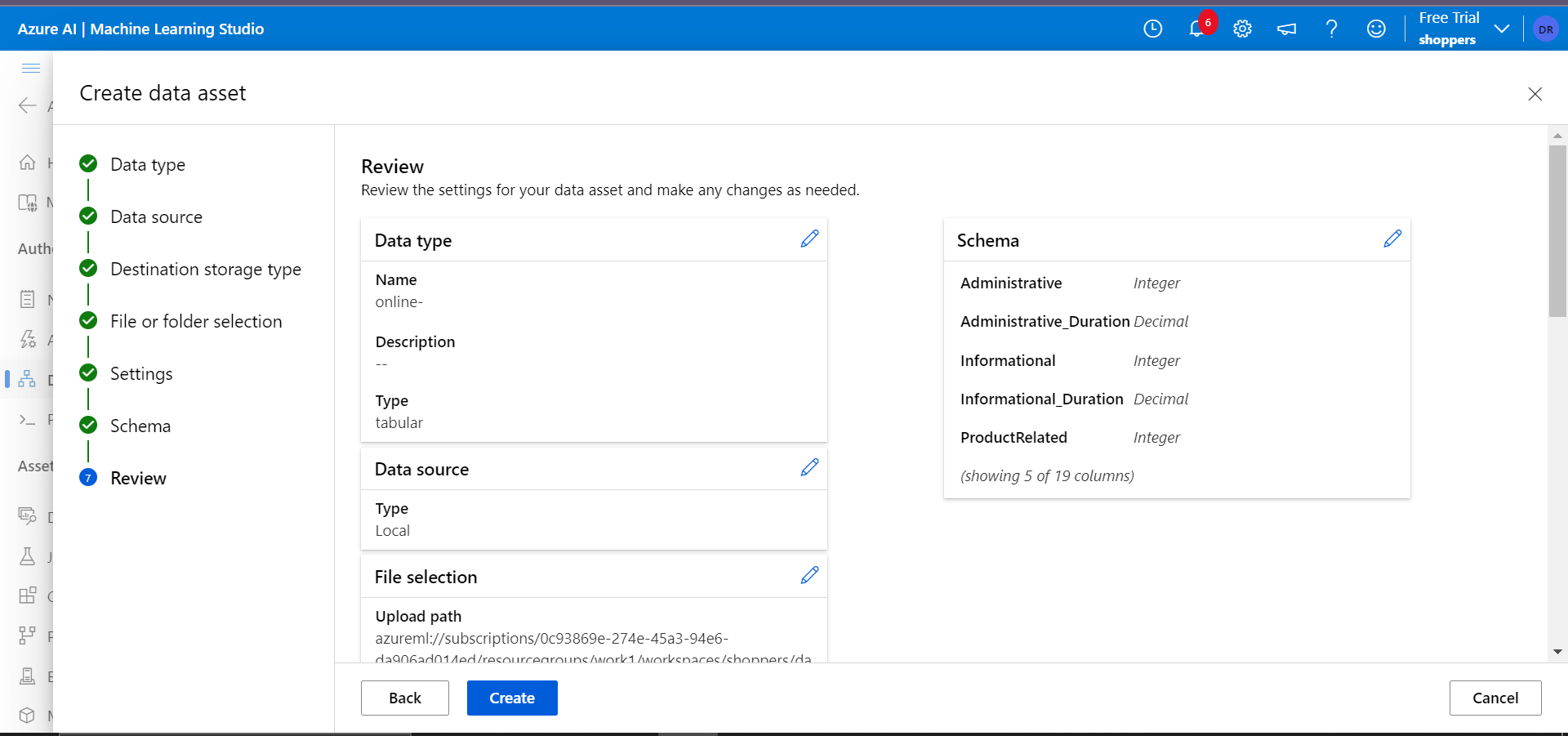


Here it will show the columns present in our data set and all the details of our data set.If we want to change anything we can change it here.

Schema:

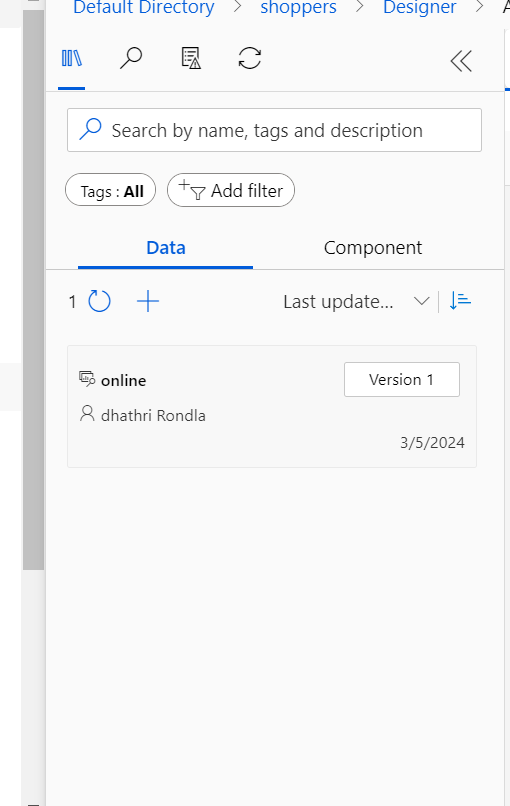


Schema step allows us to disable a column which has uniq values .



And last step is to review and click on create and data asset is created.

**Ml Pipeline Flow**



Drag the data component to the pipeline environment

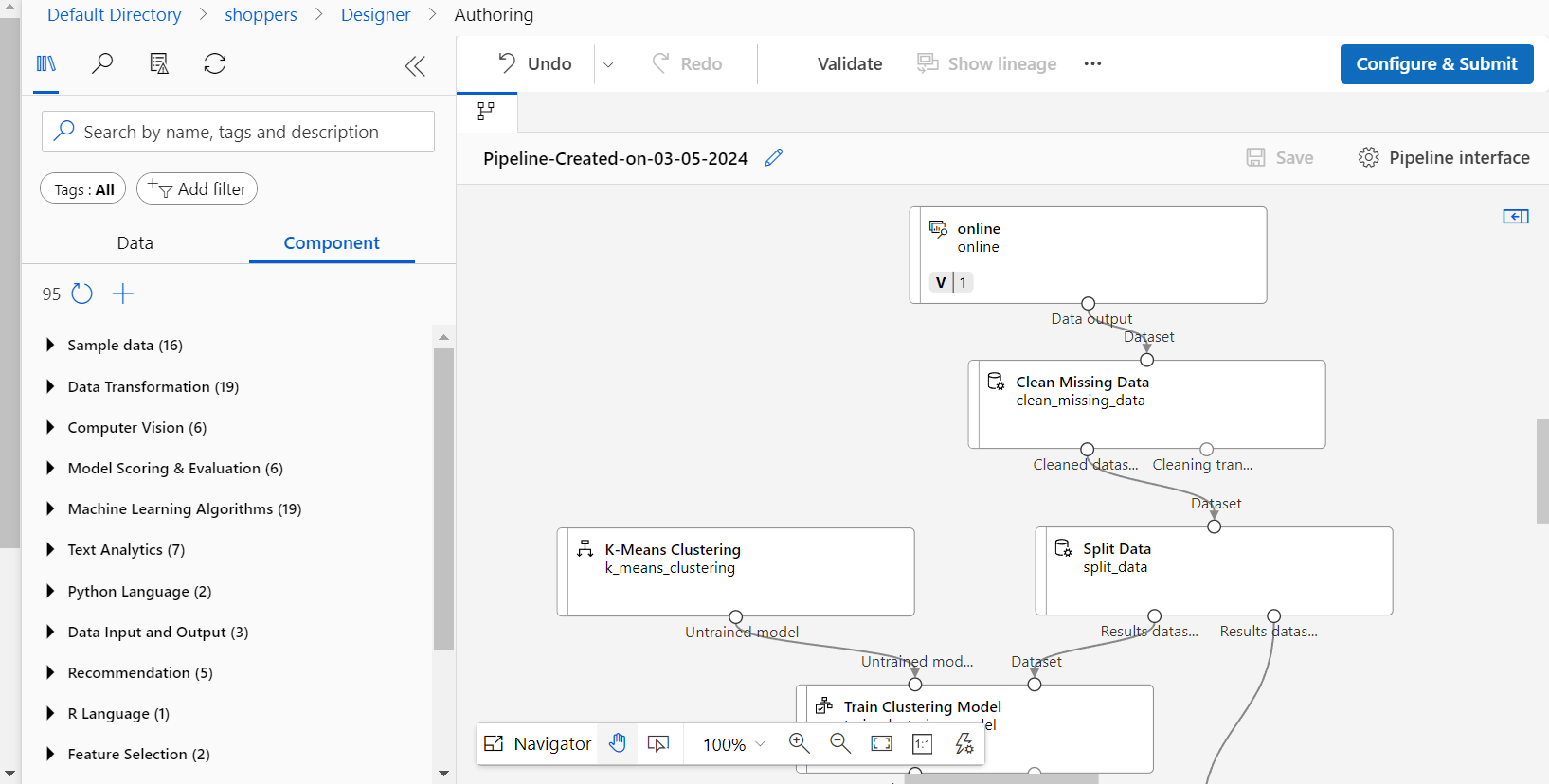


Fig :1

Here's a simplified flow of how a typical machine learning pipeline might look:

Data Ingestion:

The pipeline starts with data ingestion, where raw data is collected from various sources. This could include Azure Blob Storage, Azure SQL Database, or other data repositories.

Data Preparation:

Once the data is ingested, it undergoes preprocessing and cleaning.

For cleaning double click the **Clean Missing Data** module, and in the pane on the right, select **Edit column**. Then in the **Columns to be cleaned** window, select **With rules**, in the **Include** list select **Column names**.

For splitting the data double click on Split data and in th fraction column write the ratio in which we want to split our dataset.

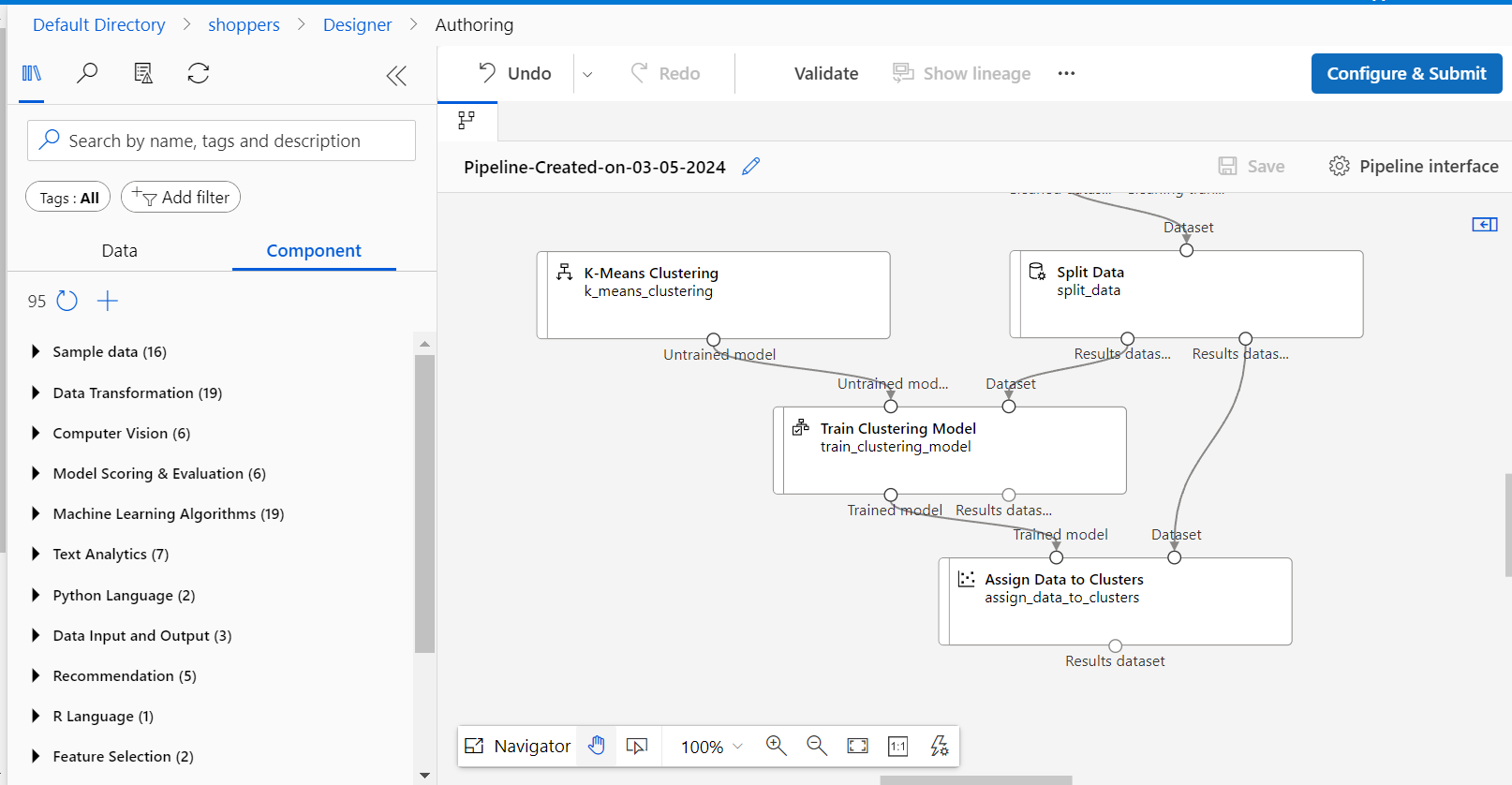


fig:1.1

1. Identifying patterns and grouping similar data points together.

2. Selecting the clusters of ( k ) to create.

3. Assigning data points to clusters by organizing and labeling them based on similarity.

4. Calculating the distance of each data point to each centroid.

5. Grouping the data points into clusters based on the calculated distances.

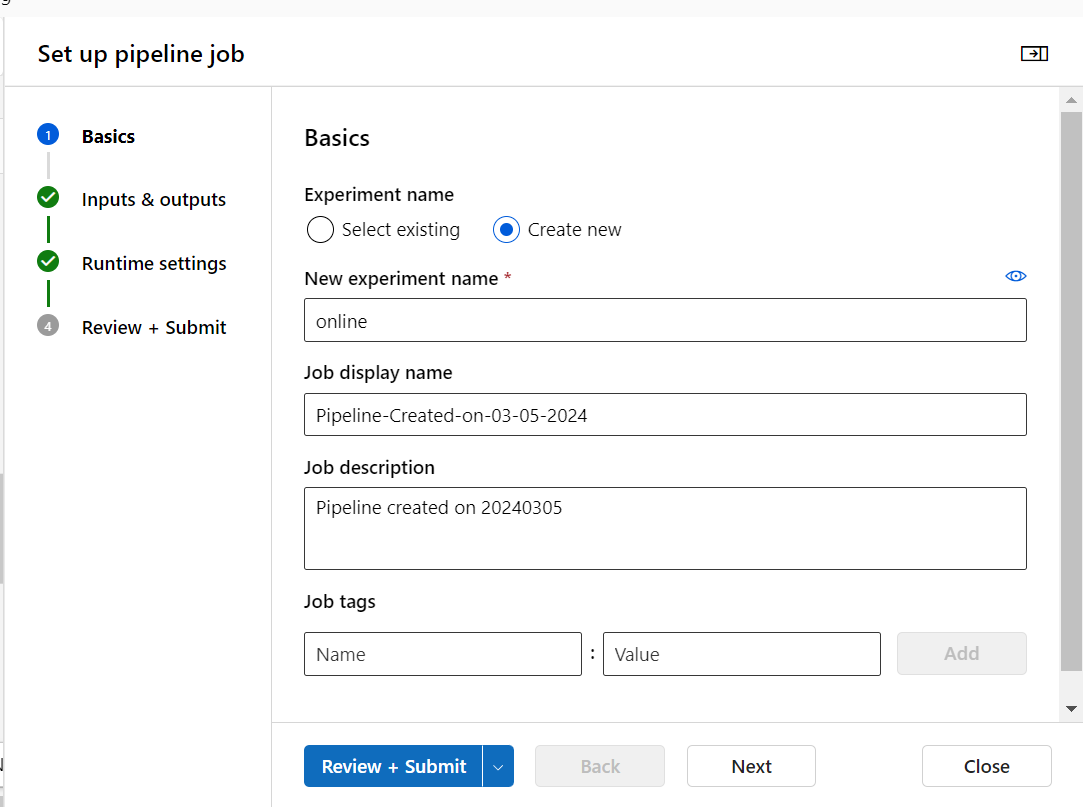


fig:1.2

Model Deployment:

SelectConfigure & Submit at the top of the page to open the Set up pipeline job dialogue.

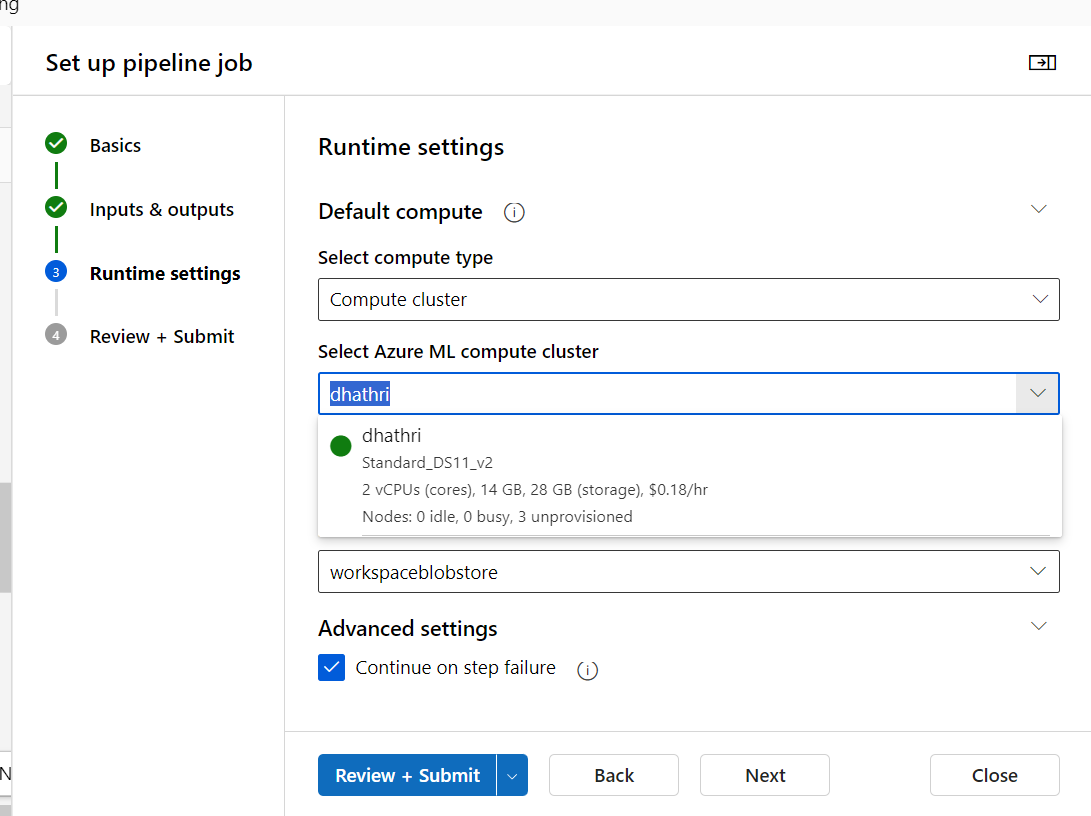
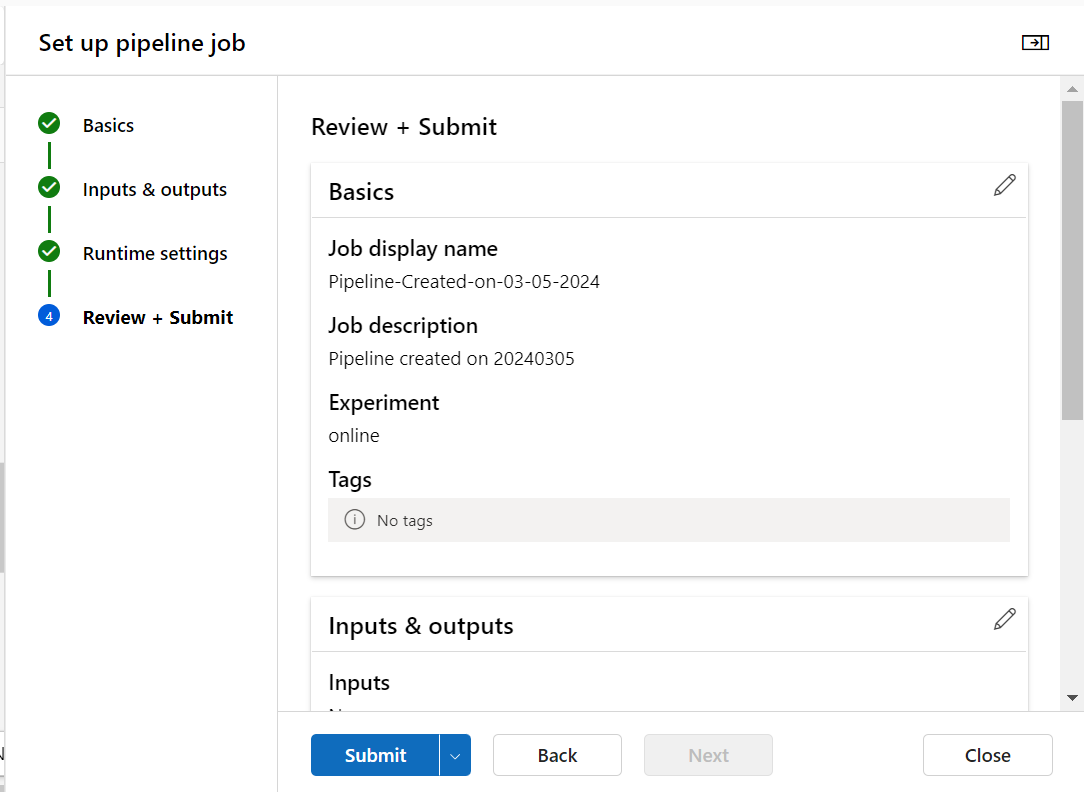


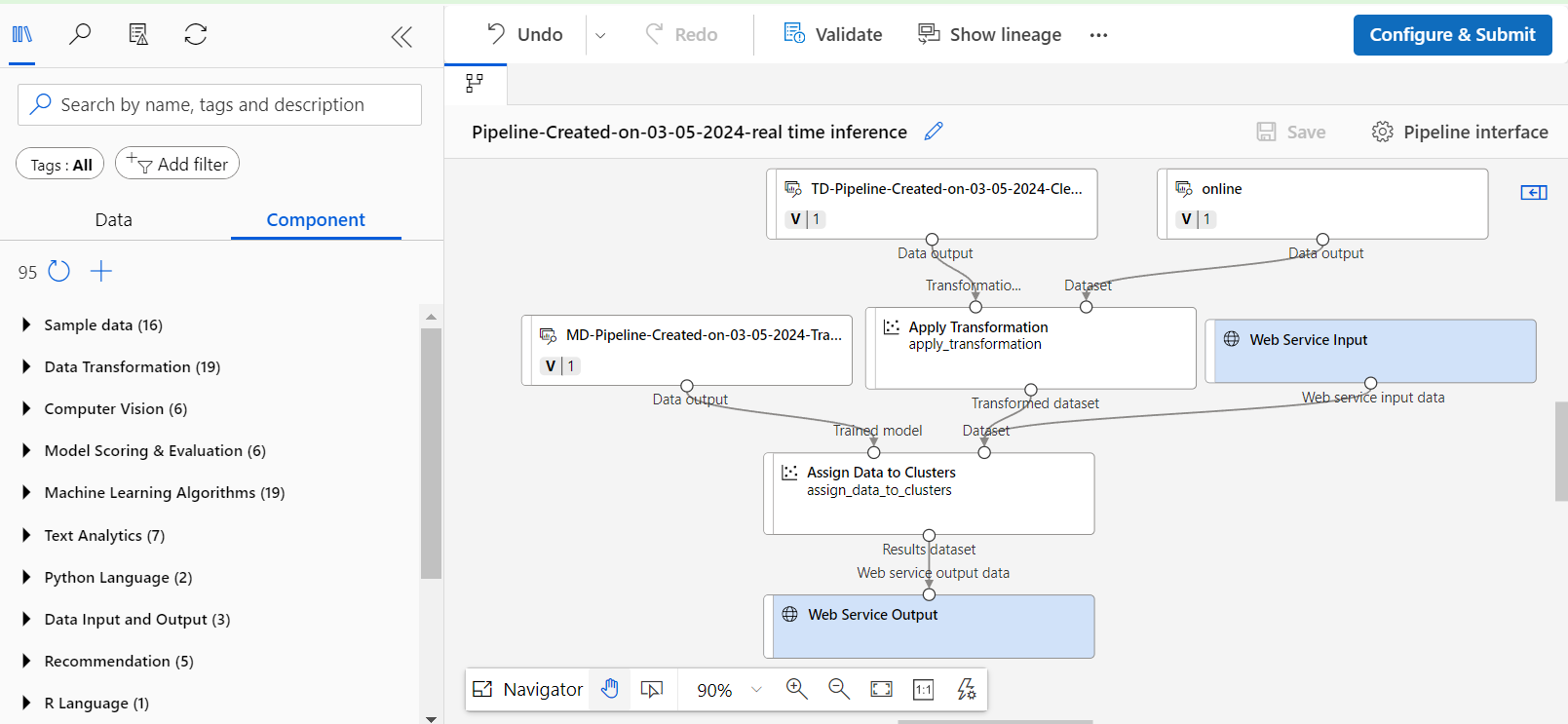
Fig:1.3

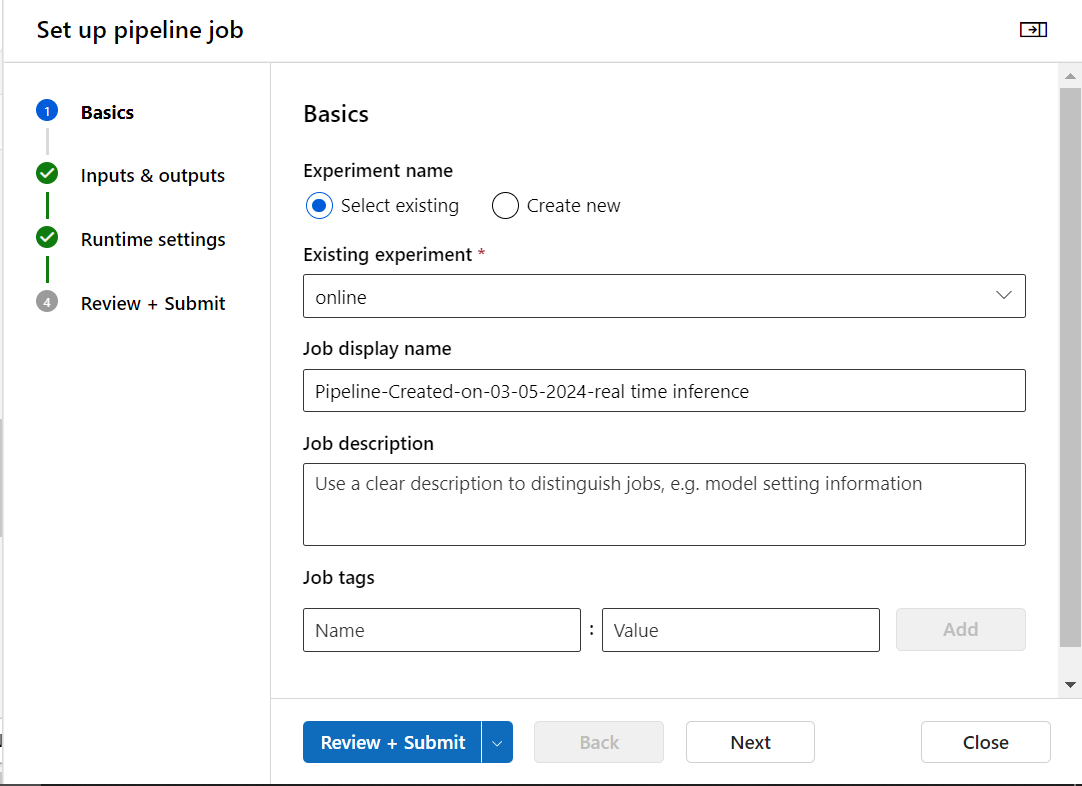
  
 fig 1.4

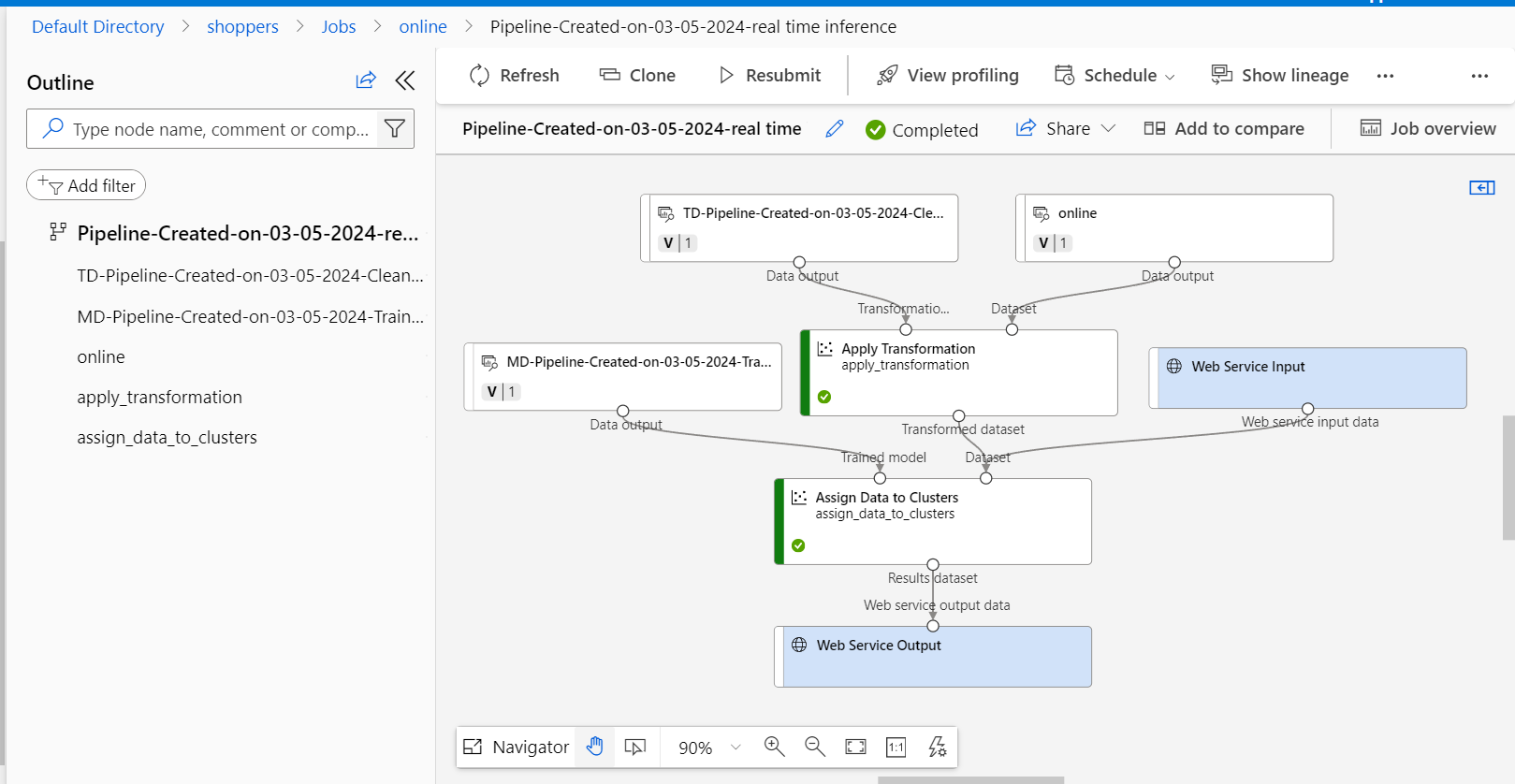
Once a satisfactory model is obtained, it can be deployed for inference.

Create the new compute instance to run the pipeline which we designed the after click on confirm submit

**Real Time Inference Pipeline:**

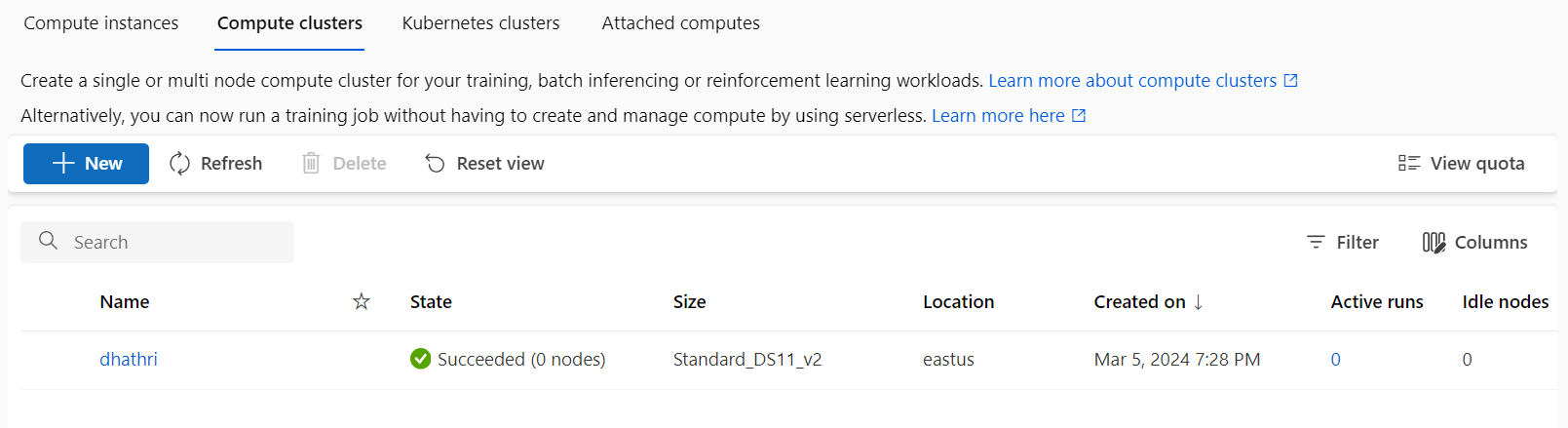


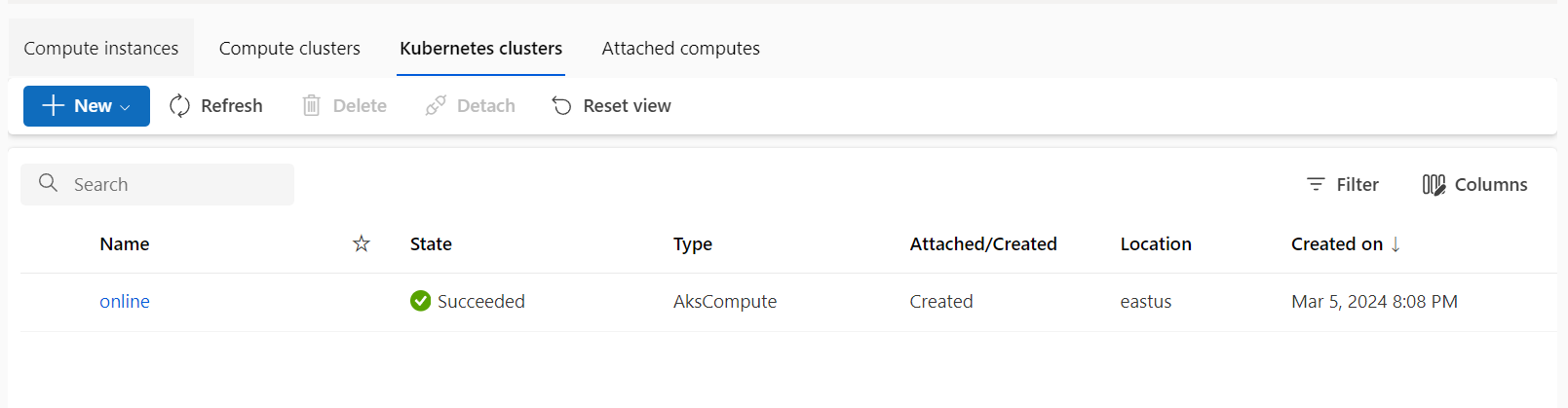




1. In the Create inference pipeline drop-down list, select **Real-time inference pipeline**. After few seconds, a new version of pipeline  will be opened.
2. Go to components and search for web service input and drag it to our model which is right side of the canvas.
3. Web service input is basically for the user to give input and Web Server Output is for getting the output.

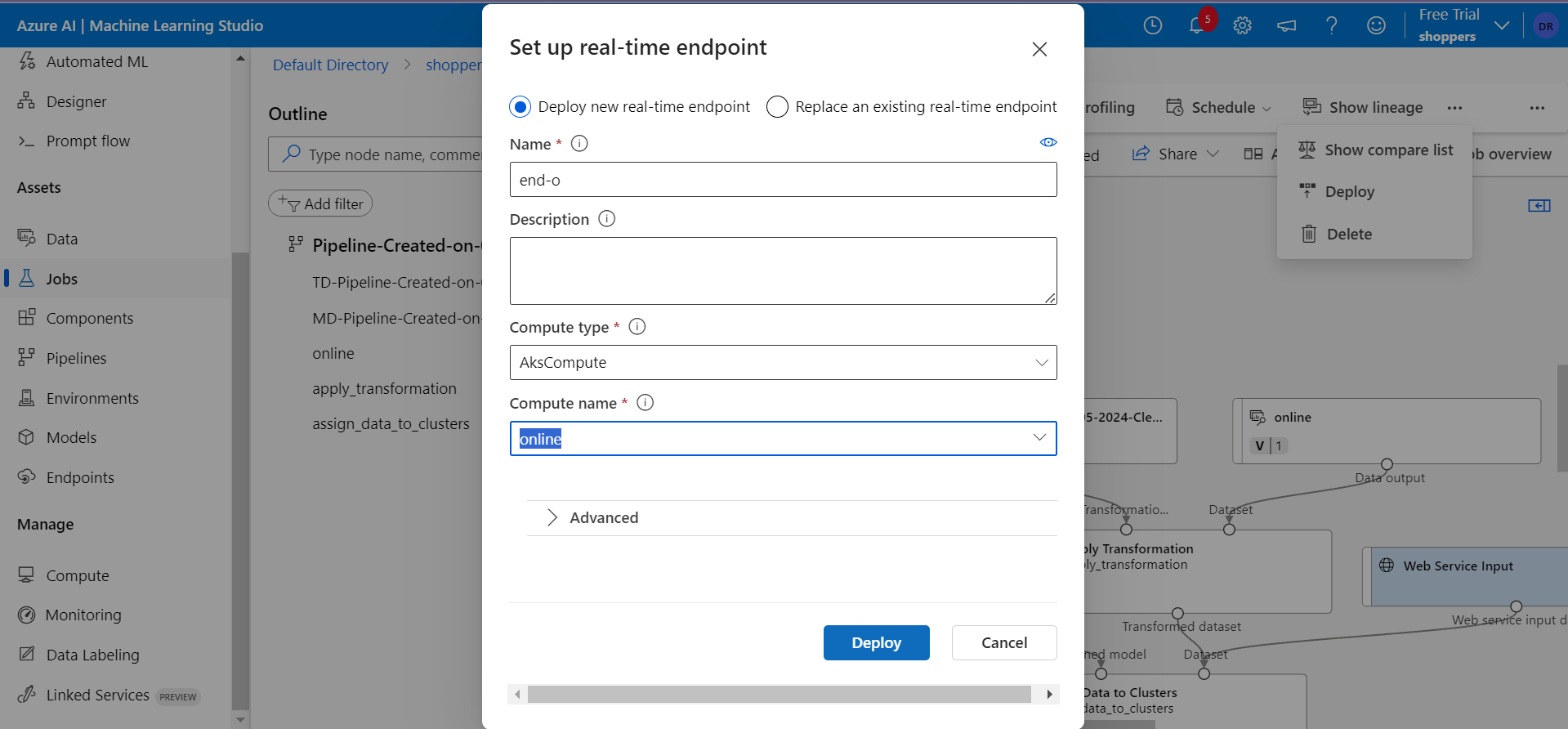
**Compute Cluster**

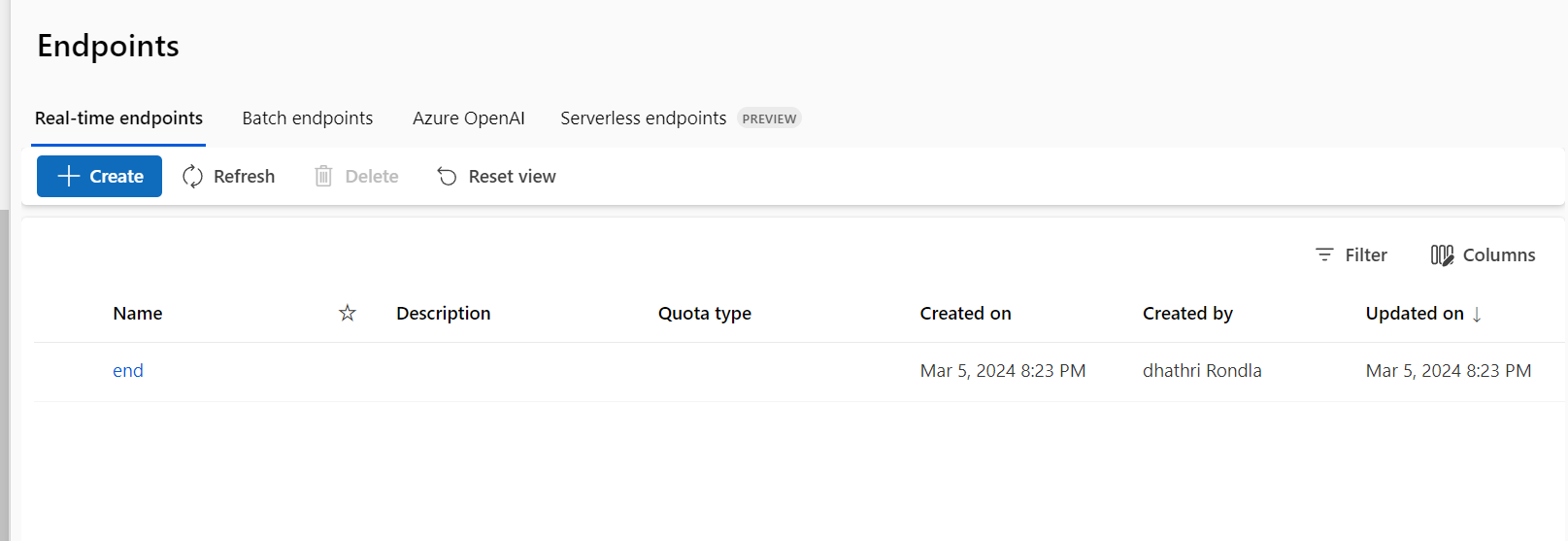


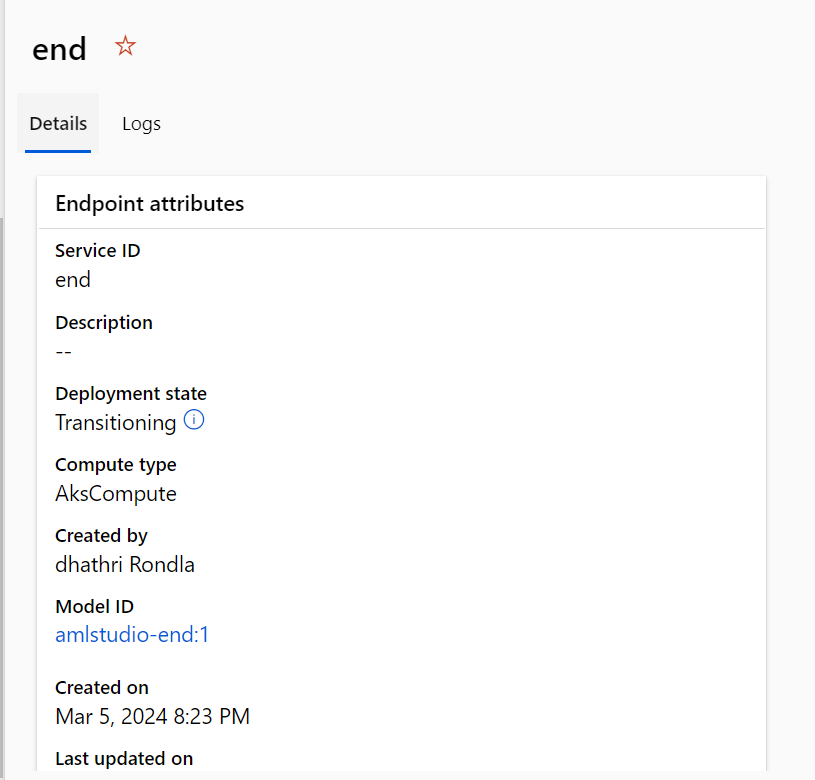
Kubernetes cluster  


After successfully deploying the real time inference,go to compute and create a  **cluster (Ask compute) In Kubemetes clusters. As we can see the above fig,it shows a Ask compute cluster called test1.**

**Endpoints**

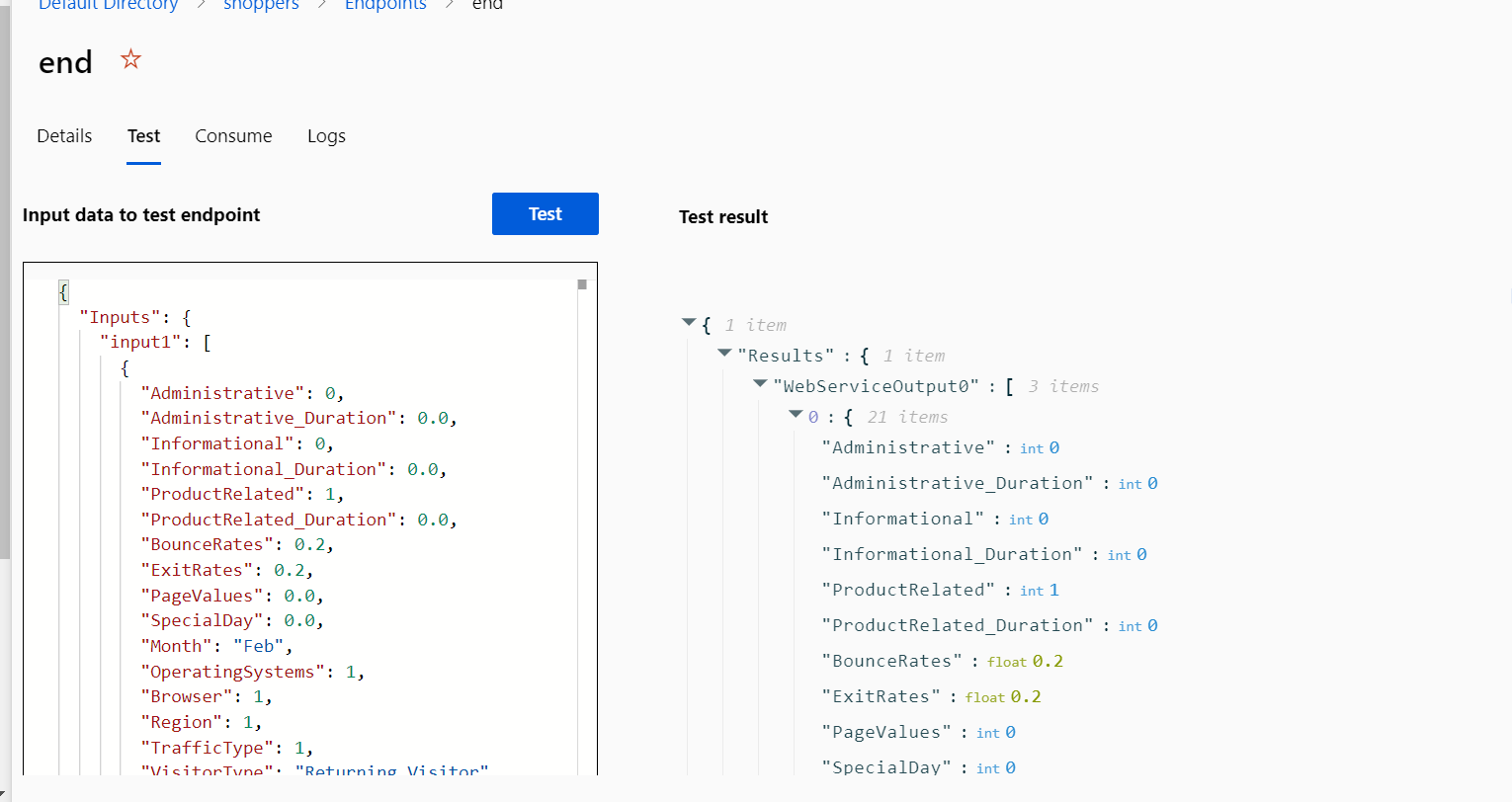






Next go to Endpoints and click on deploy and wait until the Deployment state changes to healthy state.

So after successful deployment go to the **Test** and click on test to get the output/Test result for the data in the data set.



Input:

{

  "Inputs": {

    "input1": [

      {

        "Administrative": 0,

        "Administrative\_Duration": 0.0,

        "Informational": 0,

        "Informational\_Duration": 0.0,

        "ProductRelated": 1,

        "ProductRelated\_Duration": 0.0,

        "BounceRates": 0.2,

        "ExitRates": 0.2,

        "PageValues": 0.0,

        "SpecialDay": 0.0,

        "Month": "Feb",

        "OperatingSystems": 1,

        "Browser": 1,

        "Region": 1,

        "TrafficType": 1,

        "VisitorType": "Returning\_Visitor",

        "Weekend": false,

        "Revenue": false

      },

      {

        "Administrative": 0,

        "Administrative\_Duration": 0.0,

        "Informational": 0,

        "Informational\_Duration": 0.0,

        "ProductRelated": 2,

        "ProductRelated\_Duration": 0.0,

        "BounceRates": 0.0,

        "ExitRates": 0.1,

        "PageValues": 0.0,

        "SpecialDay": 0.0,

        "Month": "Feb",

        "OperatingSystems": 2,

        "Browser": 2,

        "Region": 1,

        "TrafficType": 2,

        "VisitorType": "Returning\_Visitor",

        "Weekend": false,

        "Revenue": false

      },

      {

        "Administrative": 0,

        "Administrative\_Duration": 0.0,

        "Informational": 0,

        "Informational\_Duration": 0.0,

        "ProductRelated": 1,

        "ProductRelated\_Duration": 0.0,

        "BounceRates": 0.2,

        "ExitRates": 0.2,

        "PageValues": 0.0,

        "SpecialDay": 0.0,

        "Month": "Feb",

        "OperatingSystems": 4,

        "Browser": 1,

        "Region": 9,

        "TrafficType": 3,

        "VisitorType": "Returning\_Visitor",

        "Weekend": false,

        "Revenue": false

      }

    ]

  },

  "GlobalParameters": {}

}

Output:

{1 item

"Results":{1 item

"WebServiceOutput0":[3 items

0:{21 items

"Administrative":int0

"Administrative\_Duration":int0

"Informational":int0

"Informational\_Duration":int0

"ProductRelated":int1

"ProductRelated\_Duration":int0

"BounceRates":float0.2

"ExitRates":float0.2

"PageValues":int0

"SpecialDay":int0

"Month":string"Feb"

"OperatingSystems":int1

"Browser":int1

"Region":int1

"TrafficType":int1

"VisitorType":string"Returning\_Visitor"

"Weekend":boolfalse

"Revenue":boolfalse

"Assignments":int0

"DistancesToClusterCenter no.0":float7.382983113757291e-15

"DistancesToClusterCenter no.1":float1.0000000000000075

}

1:{21 items

"Administrative":int0

"Administrative\_Duration":int0

"Informational":int0

"Informational\_Duration":int0

"ProductRelated":int2

"ProductRelated\_Duration":int0

"BounceRates":int0

"ExitRates":float0.1

"PageValues":int0

"SpecialDay":int0

"Month":string"Feb"

"OperatingSystems":int2

"Browser":int2

"Region":int1

"TrafficType":int2

"VisitorType":string"Returning\_Visitor"

"Weekend":boolfalse

"Revenue":boolfalse

"Assignments":int0

"DistancesToClusterCenter no.0":float7.382983113757291e-15

"DistancesToClusterCenter no.1":float1.0000000000000075

}

2:{21 items

"Administrative":int0

"Administrative\_Duration":int0

"Informational":int0

"Informational\_Duration":int0

"ProductRelated":int1

"ProductRelated\_Duration":int0

"BounceRates":float0.2

"ExitRates":float0.2

"PageValues":int0

"SpecialDay":int0

"Month":string"Feb"

"OperatingSystems":int4

"Browser":int1

"Region":int9

"TrafficType":int3

"VisitorType":string"Returning\_Visitor"

"Weekend":boolfalse

"Revenue":boolfalse

"Assignments":int0

"DistancesToClusterCenter no.0":float7.382983113757291e-15

"DistancesToClusterCenter no.1":float1.0000000000000075

}

]

}

}

* + **Total Observations:**
* The samples exhibit variation in visitor behavior, with different numbers of pages visited across different categories varying bounce and exit rates, and different traffic sources.
* The output suggests that the data has been analyzed using clustering techniques. The samples have been assigned to clusters based on their similarity, with cluster centers representing typical behavior patterns.
* The distances to cluster centers indicate how well each sample fits into its assigned cluster. Samples with smaller distances are closer to the typical behavior of their cluster, while larger distances suggest some deviation.

.