AI Resource Installation Guide

Sales and Operations Planning

Last Update: January 15, 2026

TABLE OF CONTENTS

I. ABOUT

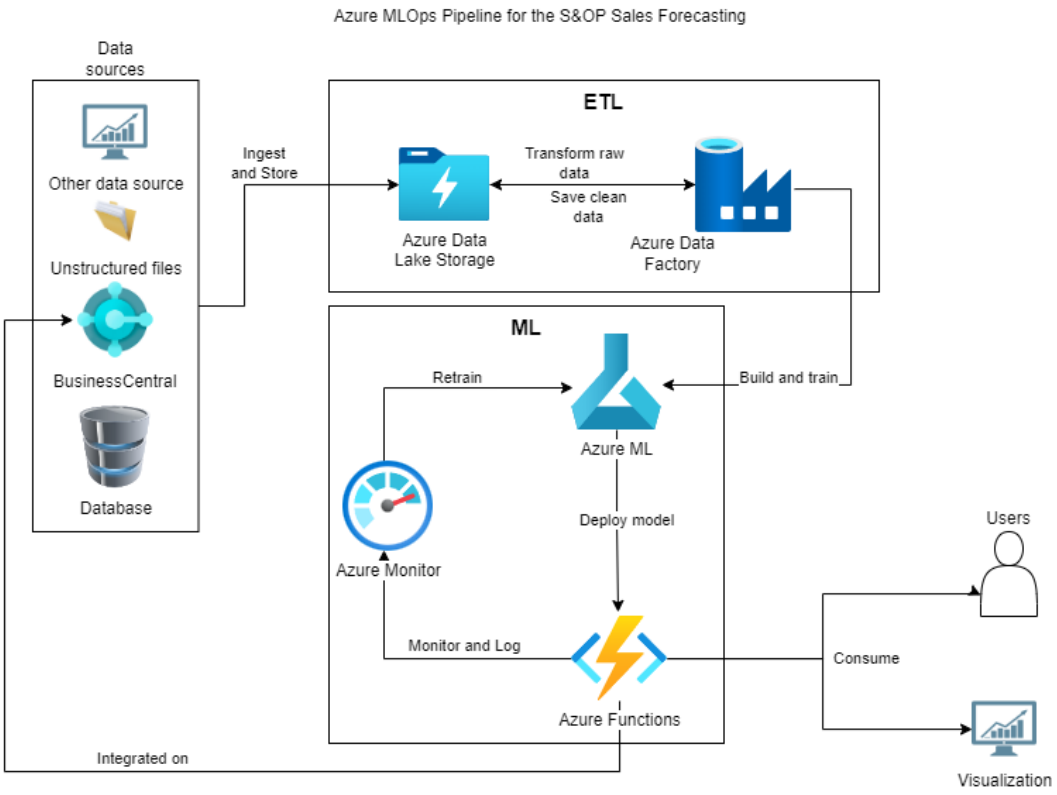
The Sales & Operations Planning for Dynamics 365 Business Central application generates AI-powered sales forecasts. Historical sales data is synced from Business Central to Microsoft Azure and then utilized for model training. The following guide covers the S&OP AI pipeline and its installation instructions.

For more information, please go to https://www.websan.com/sales-and-operations-planning-for-dynamics-365-business-central .

II. ARCHITECTURE DESIGN & PRE-REQUISITES

This section presents the Sales & Operations Planning application AI pipeline and the pre-requisites for the installation of its resources.

AI Pipeline and Data Privacy



IMPORTANT: Except for the resources marked with a double asterisk (**), the standard installation (**recommended**) requires setting resources names **exactly** to the values indicated in the present documentation. Using a different naming convention requires source code updates not supported by WebSan. If different names are required by your organization, please contact the AI Support team as soon as possible.

WebSan's AI Commitment: Your data will *never leave your Azure tenant* and will solely be used for producing forecasts for your Business Central environment. WebSan Solutions Inc. will not use your data to train the models used by other companies.

Installation Pre-Requisites

Listed below are the AI resource installation requirements:

1. An active Azure subscription that will allow the creation and usage of the following resources:
    a. Azure Data Lake Storage Gen2
    b. Azure Machine Learning Services
    c. Azure Data Factory v2
    d. Azure Function
2. As the installation process involves making Role-Based Access Control (RBAC) updates to the resources listed above, you must have an owner role or other elevated permissions, at least at the Resource Group level, for the resource group where the solution will be installed.
3. You will need to provide login credentials for an Azure account to the WebSan AI Support Team (no admin privileges are mandatory). These credentials will be used for the deployment of the Azure Function to your tenant.

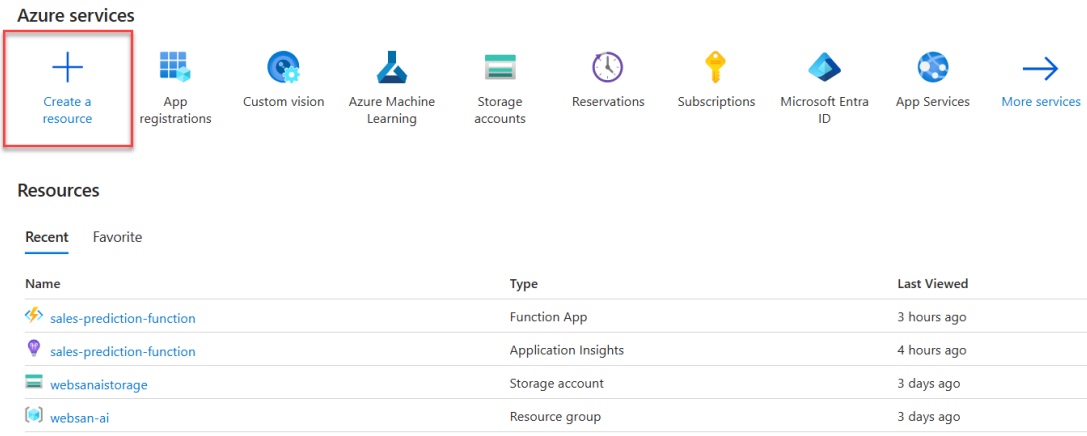
Note that you must contact the WebSan AI Support Team to gain access to the installation scripts required to complete the installation of the AI resources.

1. RESOURCE DEPLOYMENT

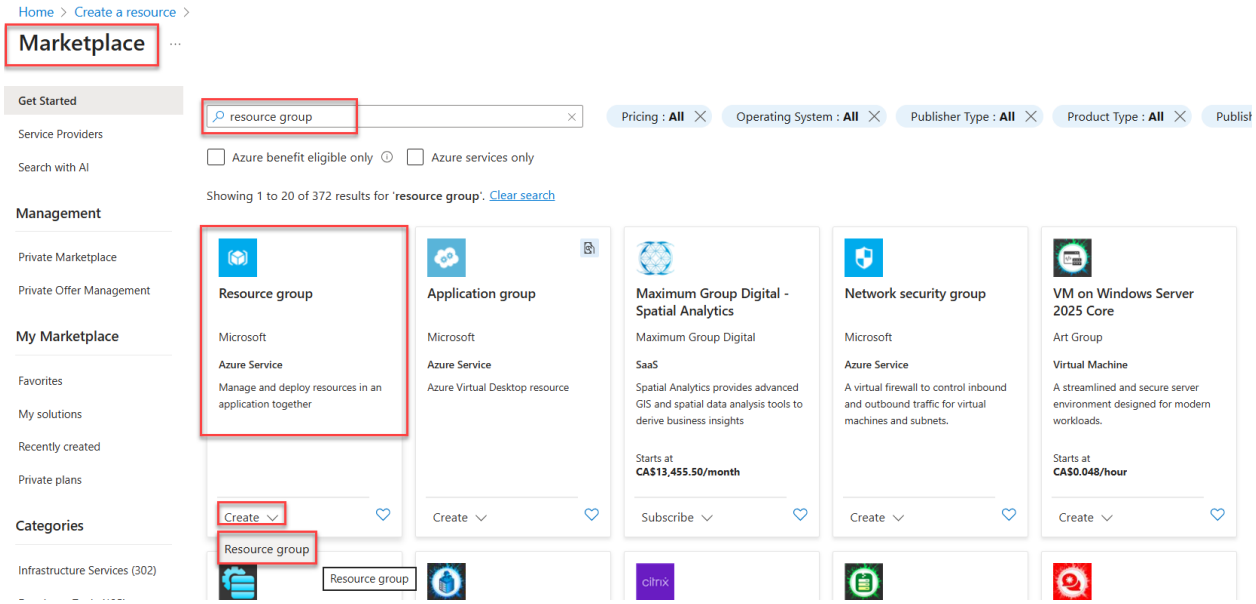
The following section covers the steps required for the deployment of the AI pipeline resources.

Resource Group

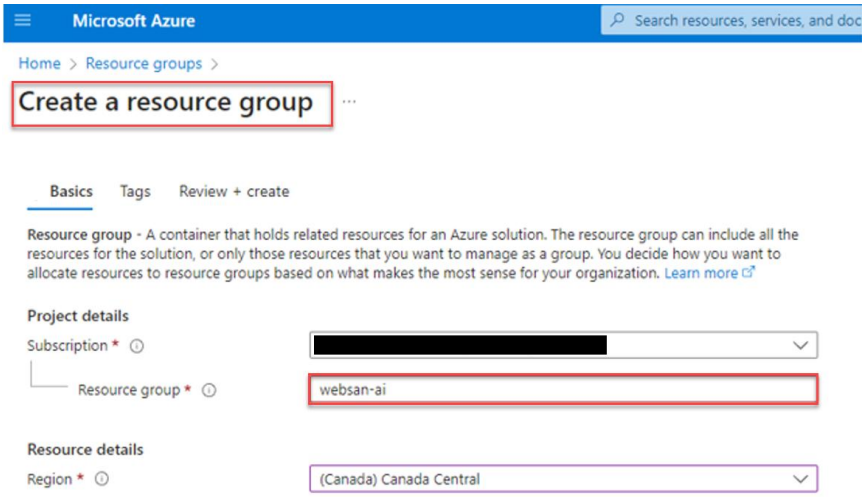
1. Log in to your Azure Portal. In the main page, click on + Create a resource.



2. In the window that appears, search for “resource group”. In the Resource group tile, click Create > Resource group.



3. In the “Create a resource group” page that appears, select your subscription and enter the name **websan-ai** in the Resource group field. Set the Region field to your desired region.



4. Click **Review + create.**

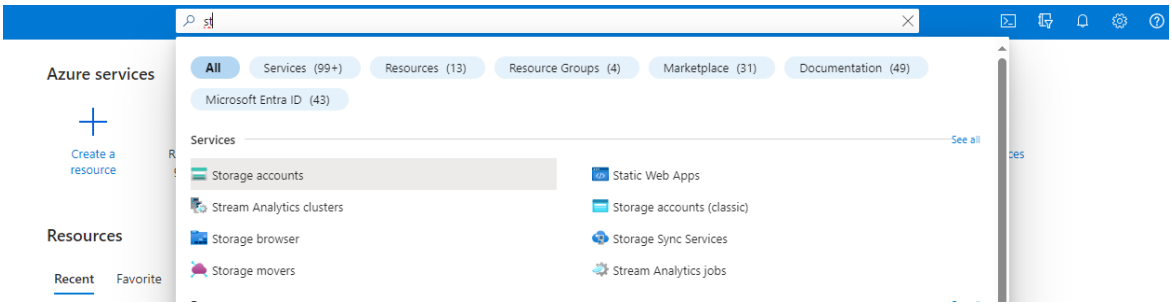
Note: You must ensure all subsequent resources created as part of this solution are located in the Resource Group you just created. Also, as you will need to make Role-Based Access Control updates to the resources that will be created next, you must have an Owner role or other elevated permissions, at least at the Resource Group level, for the resource group you just created.

Azure Data Lake Storage (ADLS)

(**) – the name of this *resource* must be globally unique across all Azure Tenants; you may set it to your preferred value. However, note that the name of the *containers* within the ADLS must follow the exact names indicated in this documentation.

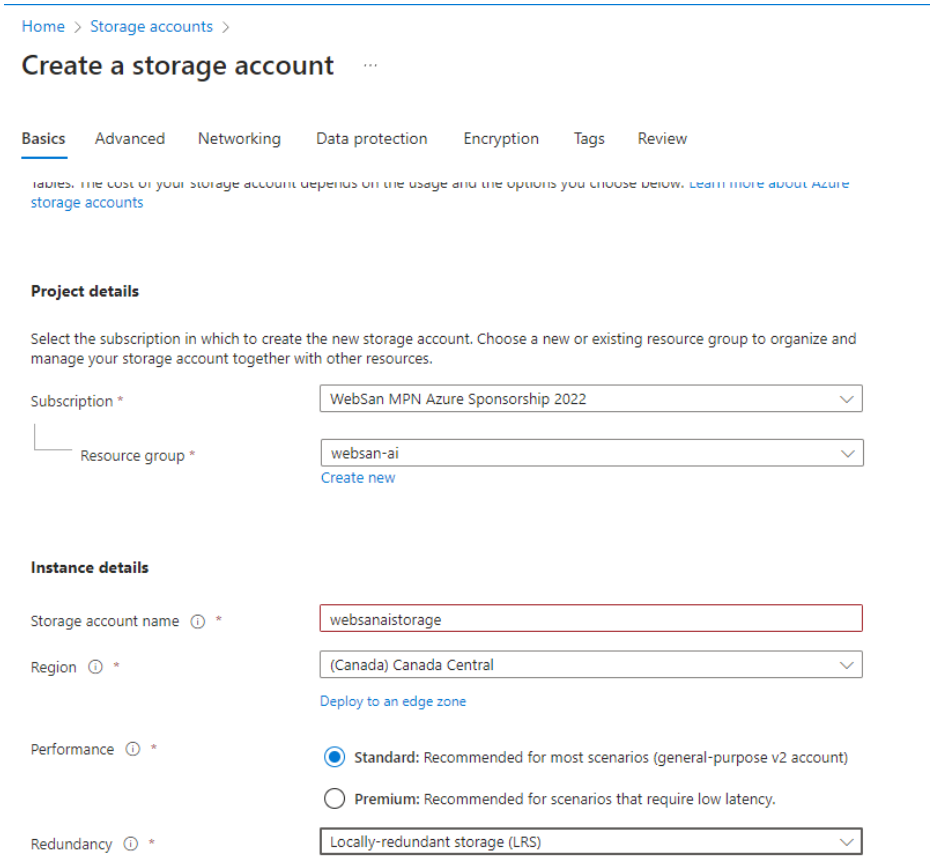
1. To create the Azure Data Lake Storage Gen2, you will need to:

    a. Search for 'Storage Accounts', click the **Storage accounts** service.

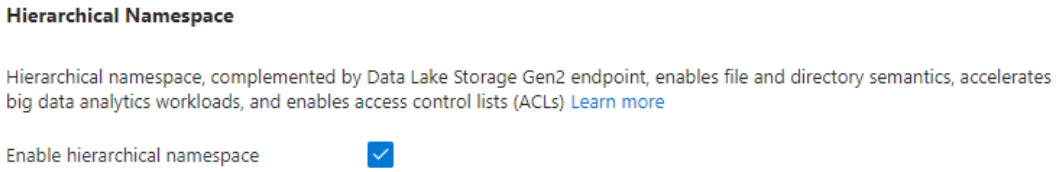


    b. Create a storage account by clicking the Create button; you will be directed to the wizard to create the storage. Set the Storage Account Name and Region to your

preferred values. **Redundancy** must be set to **Locally-redundant storage (LRS)**. Make sure to choose the resource group you created before.



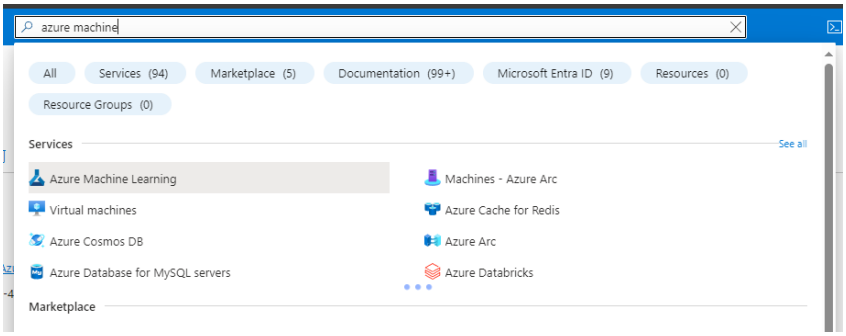
c. On the Advanced tab, look for Hierarchical Namespace, then enable it.



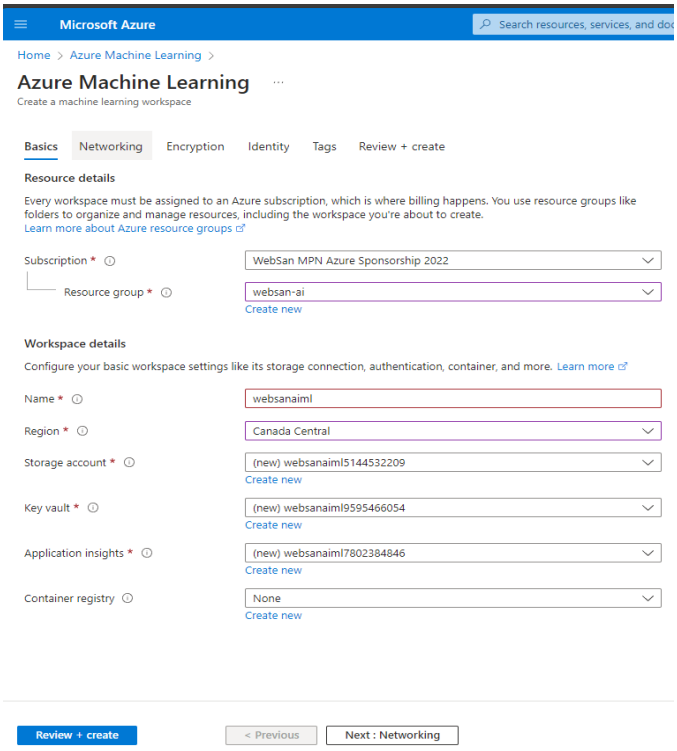
d. Click the **Review + Create > Create** buttons and wait for it to be deployed.

Azure Machine Learning

1. To create the next resource – **Azure Machine Learning Services:**
    a. On your Azure Portal, search for 'Machine Learning', then click the **Azure Machine Learning** services.



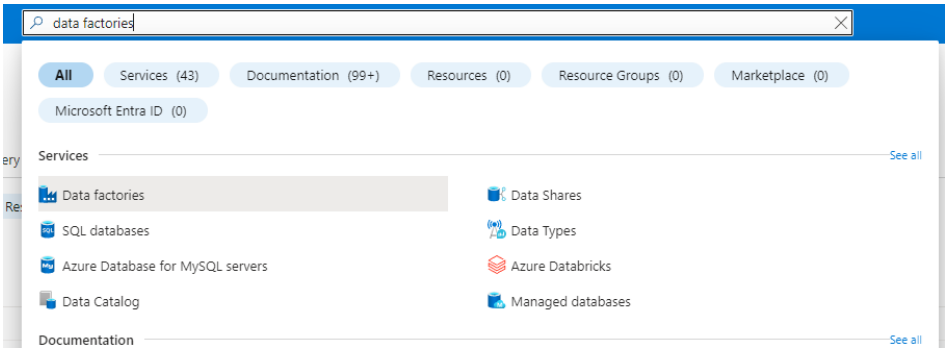
b. Create a machine learning workspace by clicking the Create button. It shall direct you to the wizard for creating the workspace. Name the workspace **websanaiml,** then set the Region based on your preference. Values for the storage account, key vault, and application insights will be automatically created.



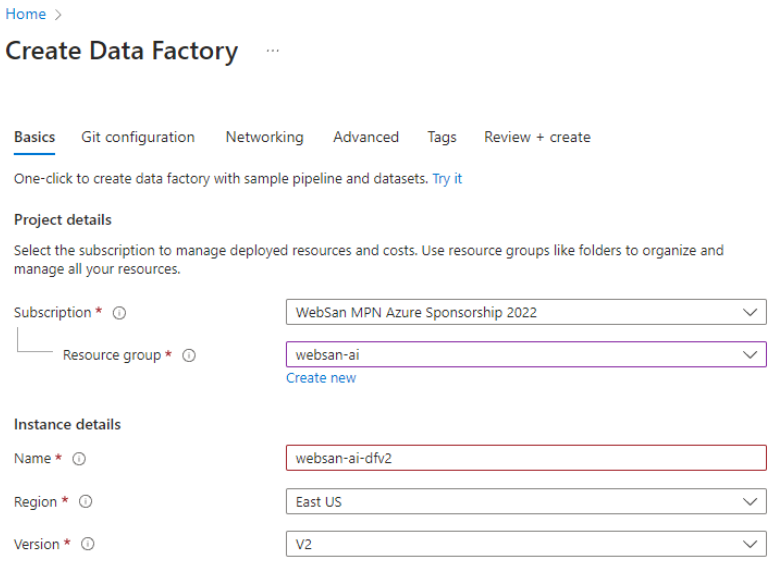
c. Click the **Review + create > Create** buttons. Wait for it to be deployed.

Azure Data Factory

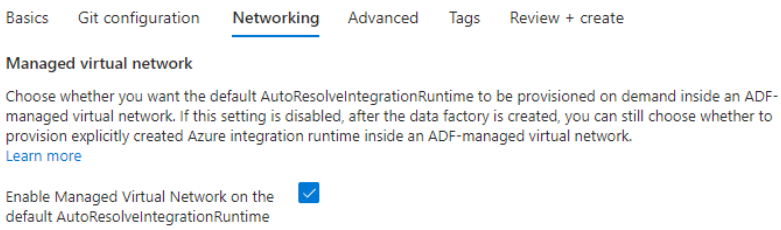
1. To create the next resource – **Azure Data Factory v2**:
    a. On your Azure portal, search for “data factories”, then click the **Data factories** services.



b. Create the data factory by clicking the Create button. It shall direct you to the wizard for creating the data factory. Name the data factory **websan-ai-dfv2**, then set the Region based on your preference. Also, set the Version to **V2**.



c. On the Networking tab, look for Managed virtual network, then enable the option.



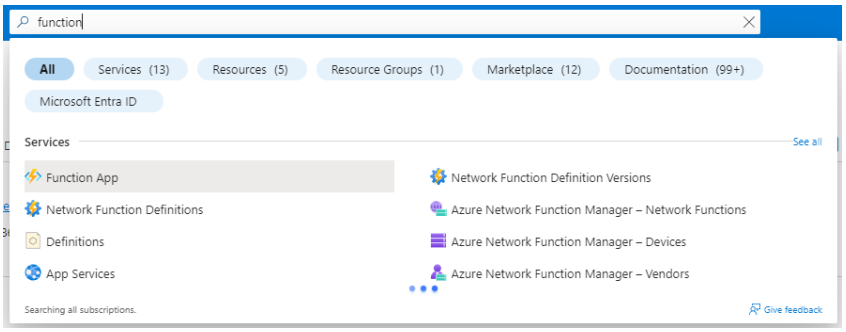
d. Click the **Review + create** > **Create** buttons. Wait for it to be deployed.

Azure Function App

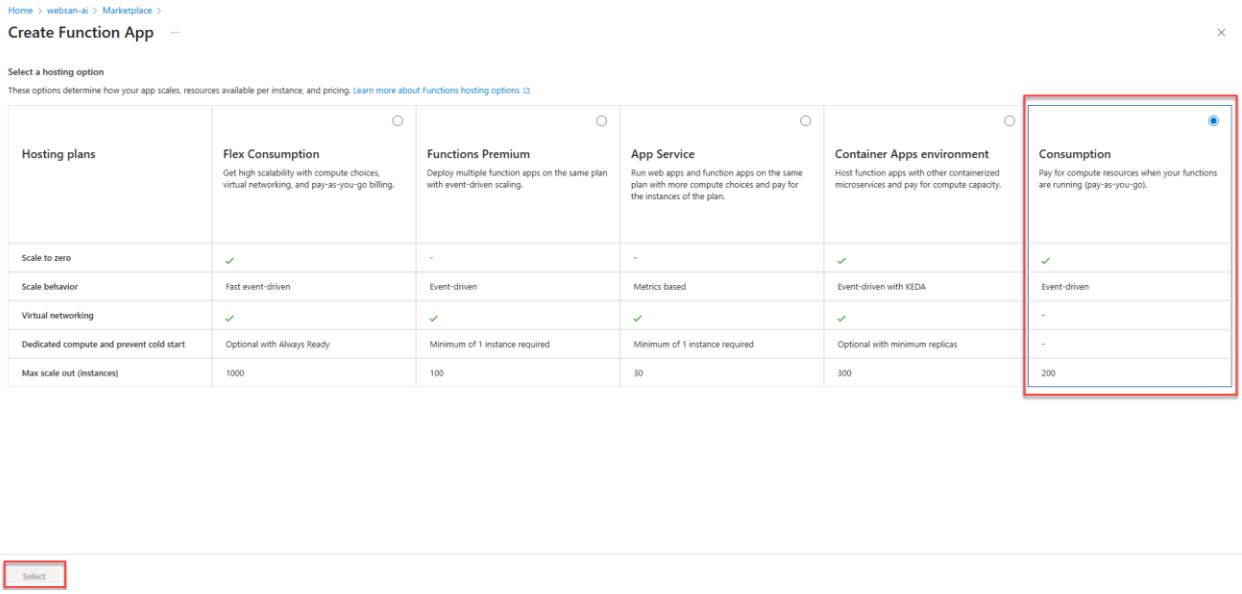
(**) – the name of this resource must be globally unique across all Azure Tenants; you may set it to your preferred value.

Install the last resource, **Azure Functions**:

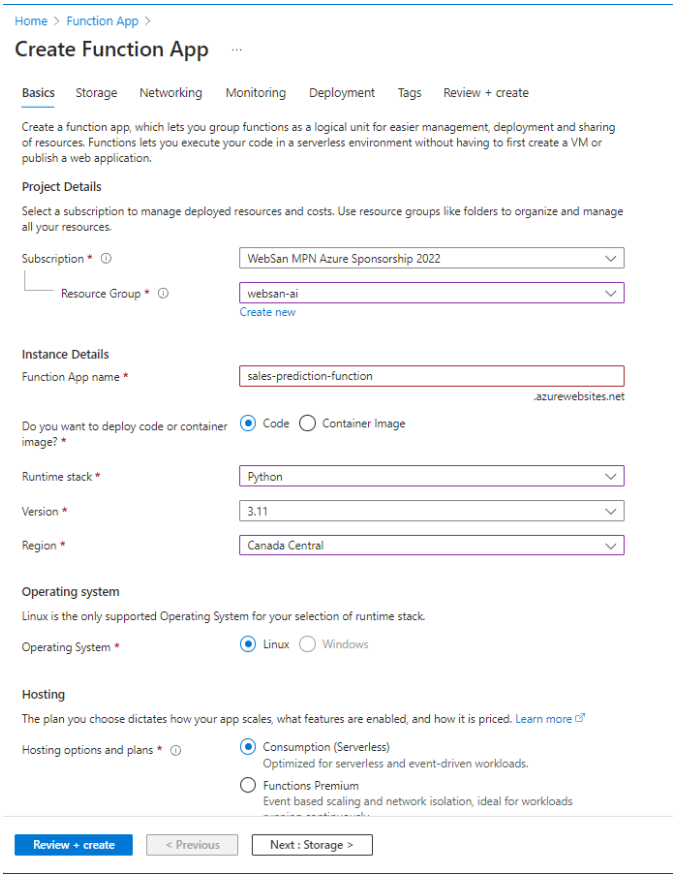
a. On your Azure portal, search for “function apps”, then click the Function App



b. In the window that appears, select the Consumption tile, accept the notifications that appear to proceed, and then click select.

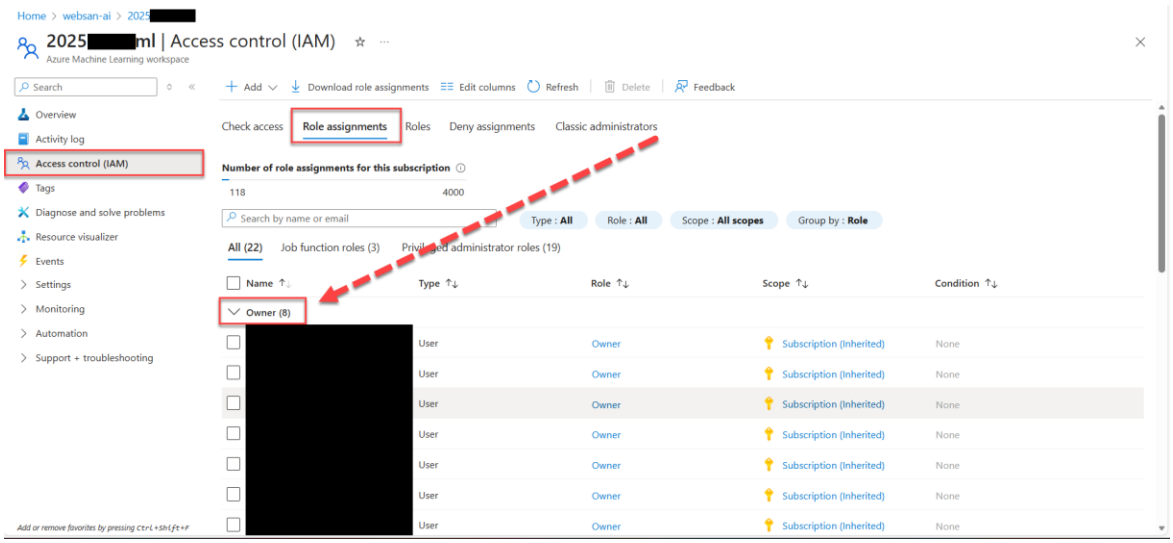


c. Enter a globally unique name in the Function App Name field, then set the following values:



d. Click the **Review + create** > **Create** buttons. Wait for it to be deployed.

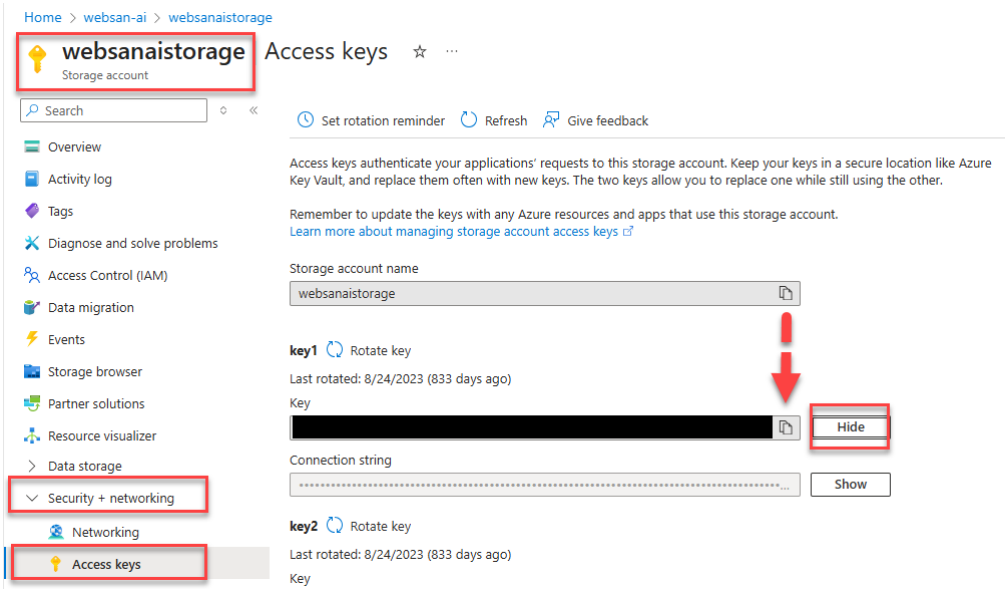
After installing all the necessary resources for the solution, ensure your Azure account is set as **Owner** for all the resources installed. You can verify this by going to the **Access Controls (IAM)** tab for each of the resources. Remember that all the resources must have been installed in the Resource Group you created at the beginning of this section.



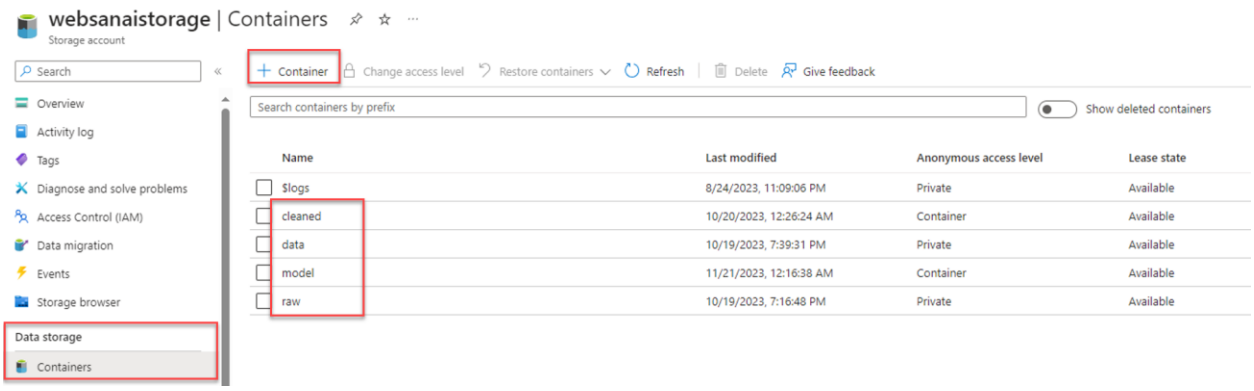
2. RESOURCE CONFIGURATION

Azure Data Lake Storage

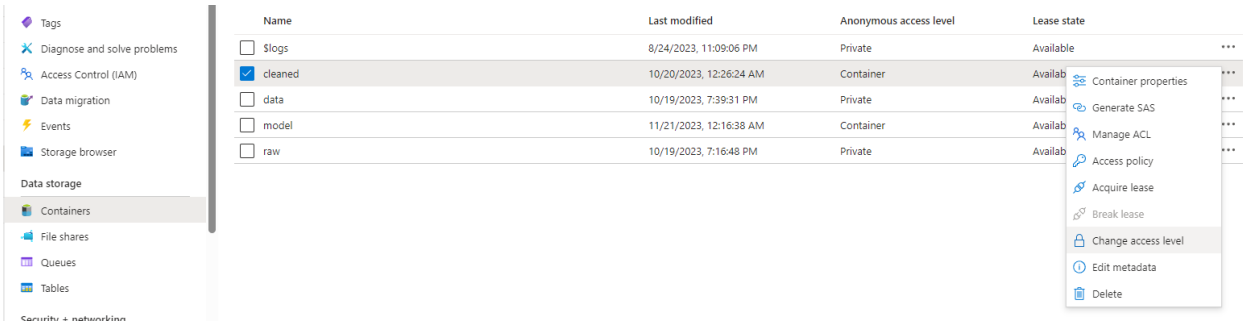
1. Go to your websan-ai resource group and select your Storage Account resource.
2. On the right pane, go to Security + networking > Access Keys. For either key, click the Show button that appears next to it, copy the key and **temporarily** store it in a safe location. You will use this key shortly.



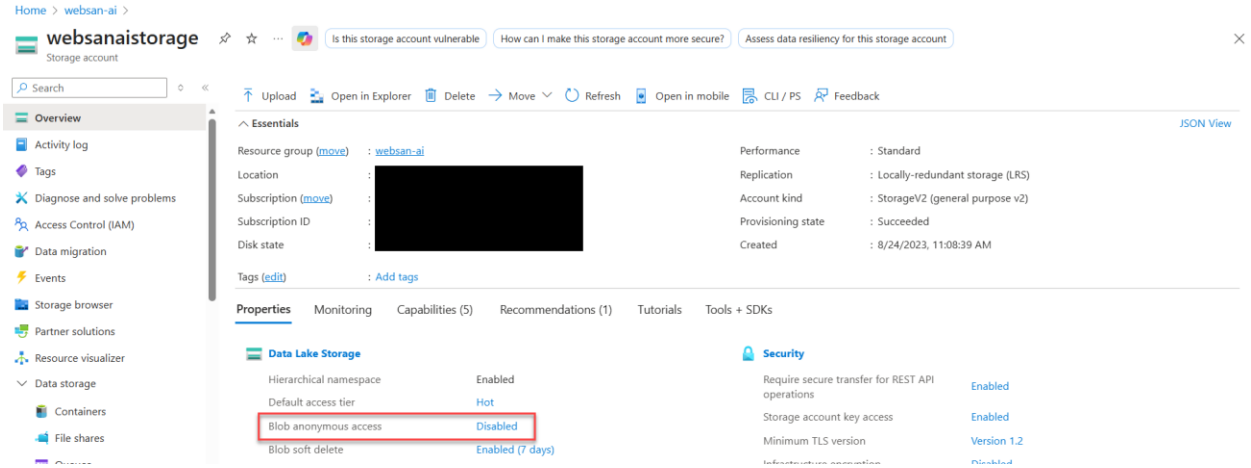
3. Next, on the right pane, go to Data storage > Containers. Here, click on the +Add container button that appears on the top-left corner of the screen. This will allow you to create the following containers in the storage account (see image below). These containers will serve as our central storage for the raw data, cleaned data (from Azure Data Factory v2), and the trained model (for Azure Function).



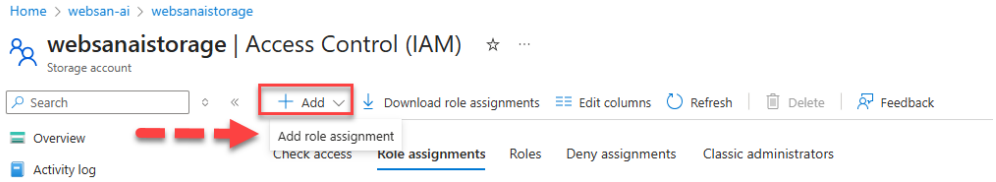
4. Confirm that the Anonymous access level is set to "Private". If necessary, change the access level by clicking the 3 horizontal dots at the end of each folder, then click the **Change access level** menu.



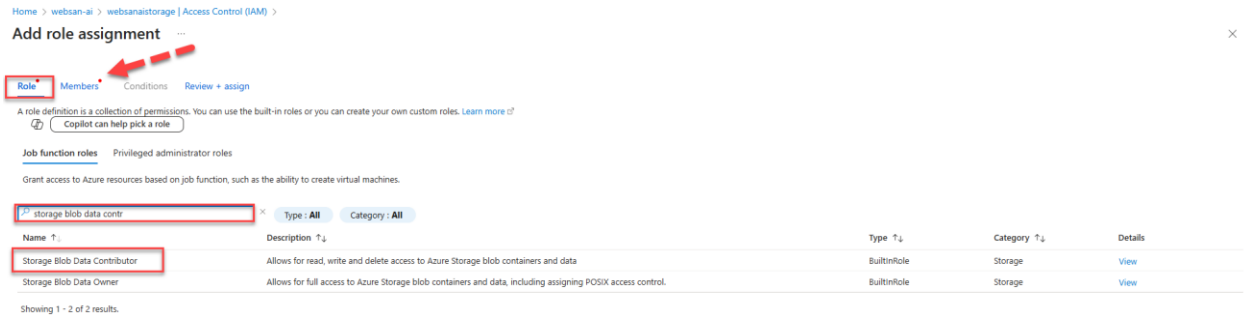
Blob anonymous access should be set to **Disabled**.



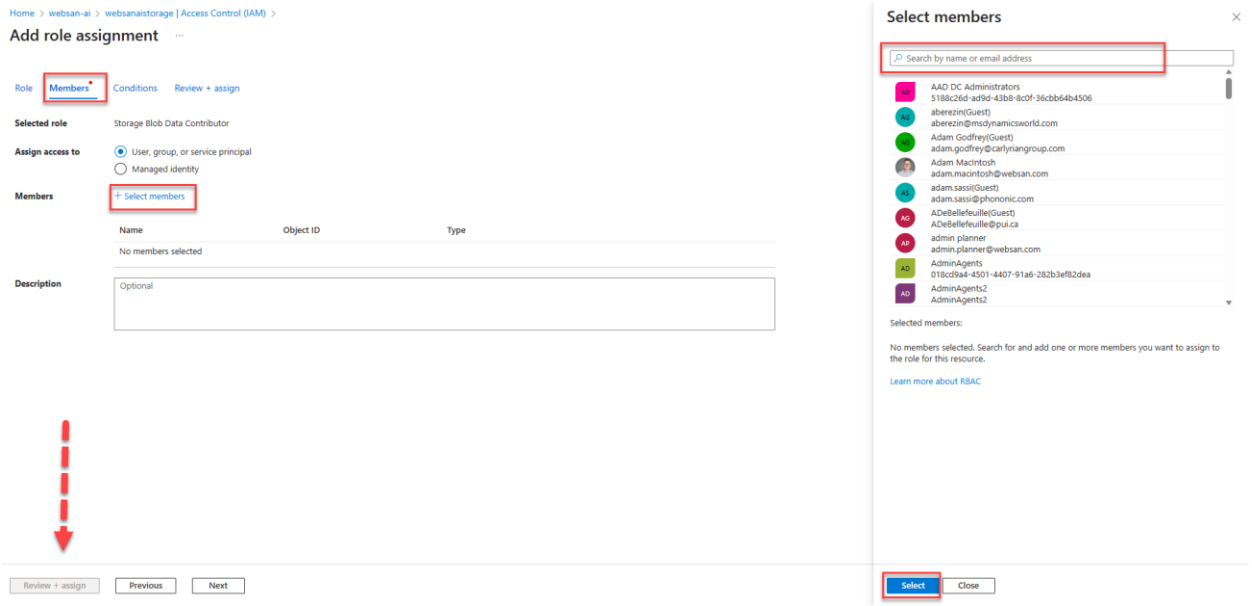
5. Click the **Access Control (IAM)** menu in the sidebar, click the + Add button -> Add role assignment.



6. In the window that appears, search for “Storage blob data contributor” and select it from the list; then, go to the Members tab.

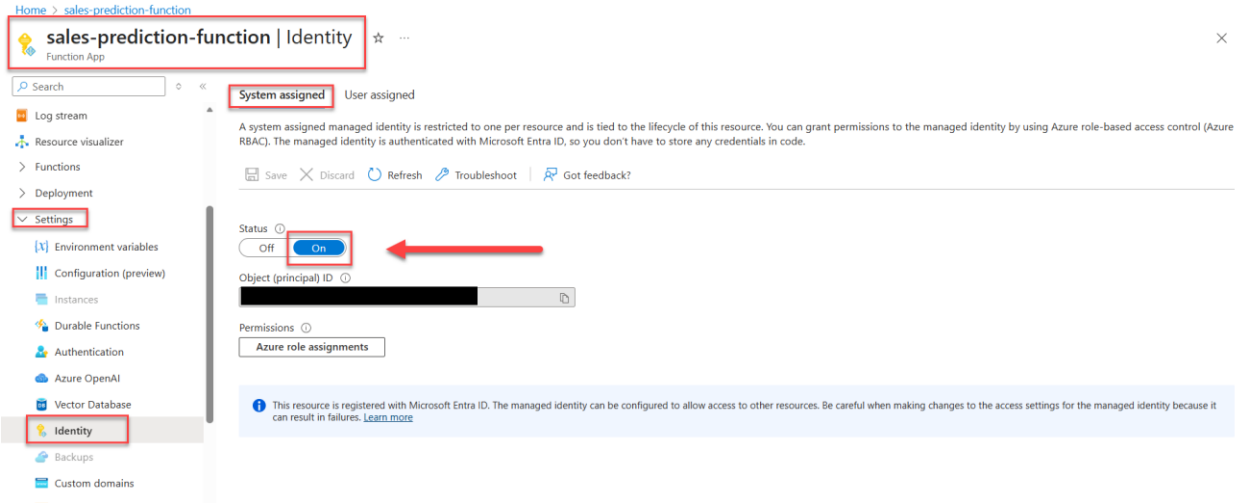


7. In the members tab, click the "+ Select members button. In the pane that appears, select your service account; then, click Review + assign.

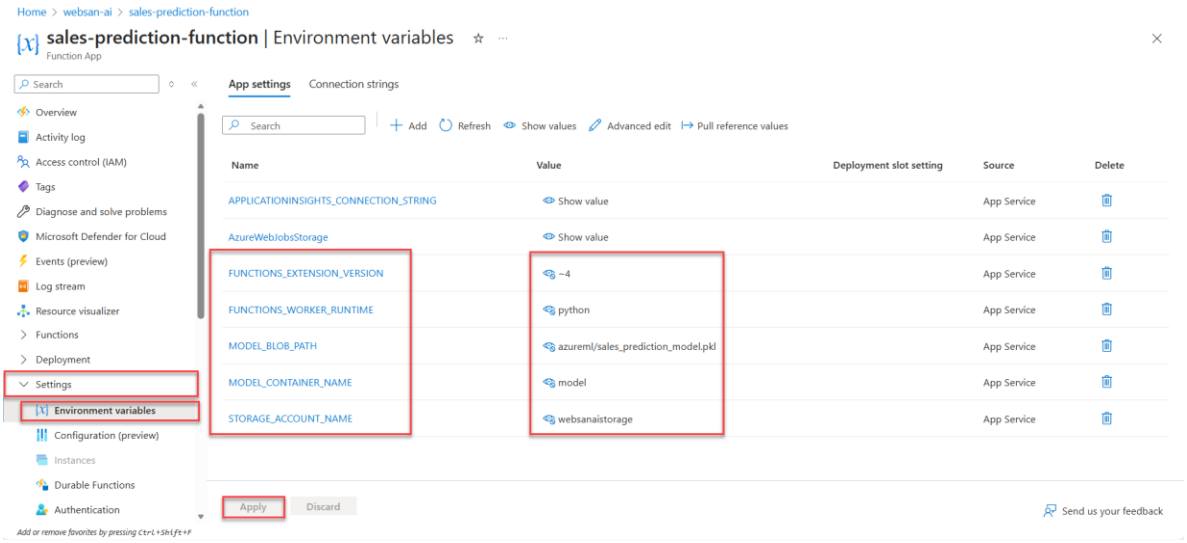


Azure Function (i)

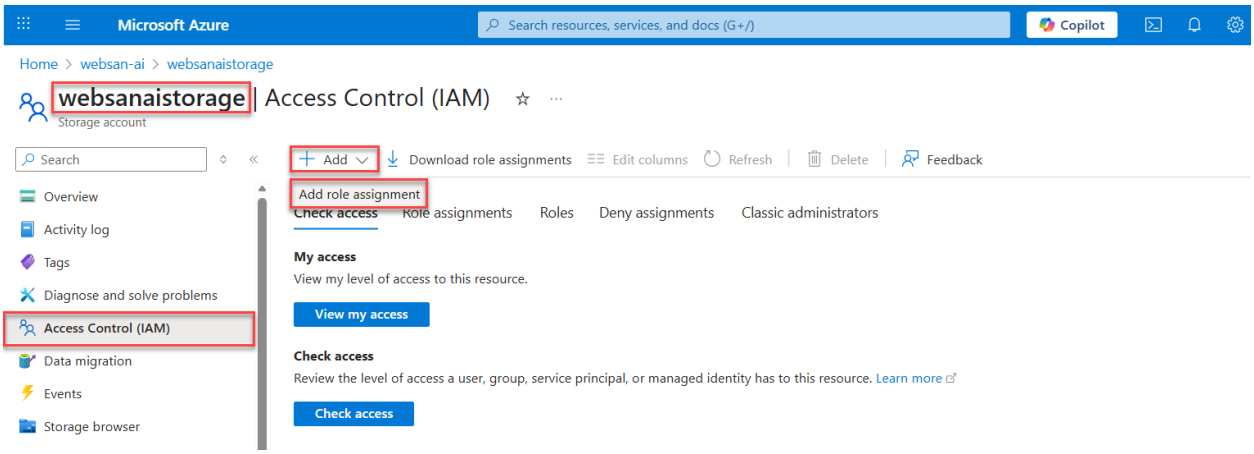
1. Go to your websan-ai resource group and select your Function App resource.
2. In the left pane, go to Settings > Identity. Under System assigned, locate the field Status and make sure it is set to "On".



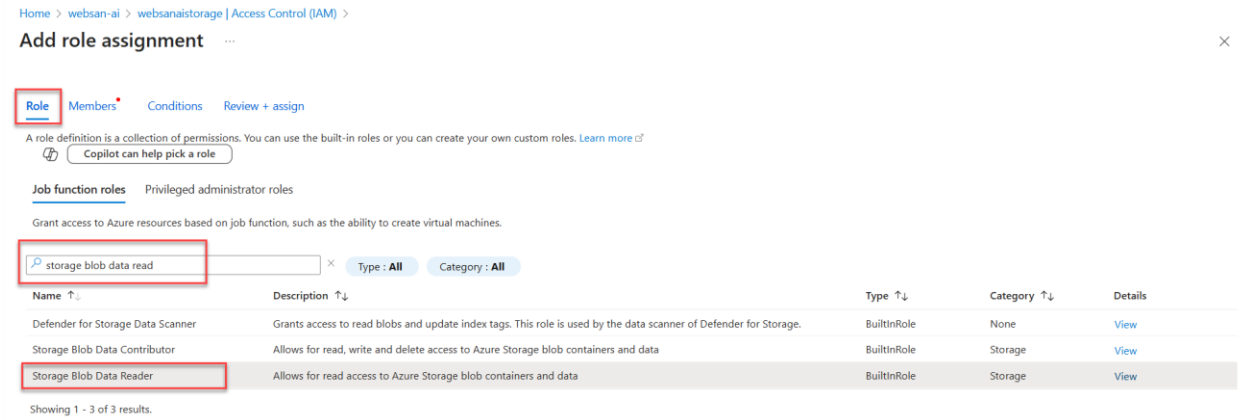
3. Go to Settings > Environment variables. Under App Settings, create the environment variables shown below and click Apply.



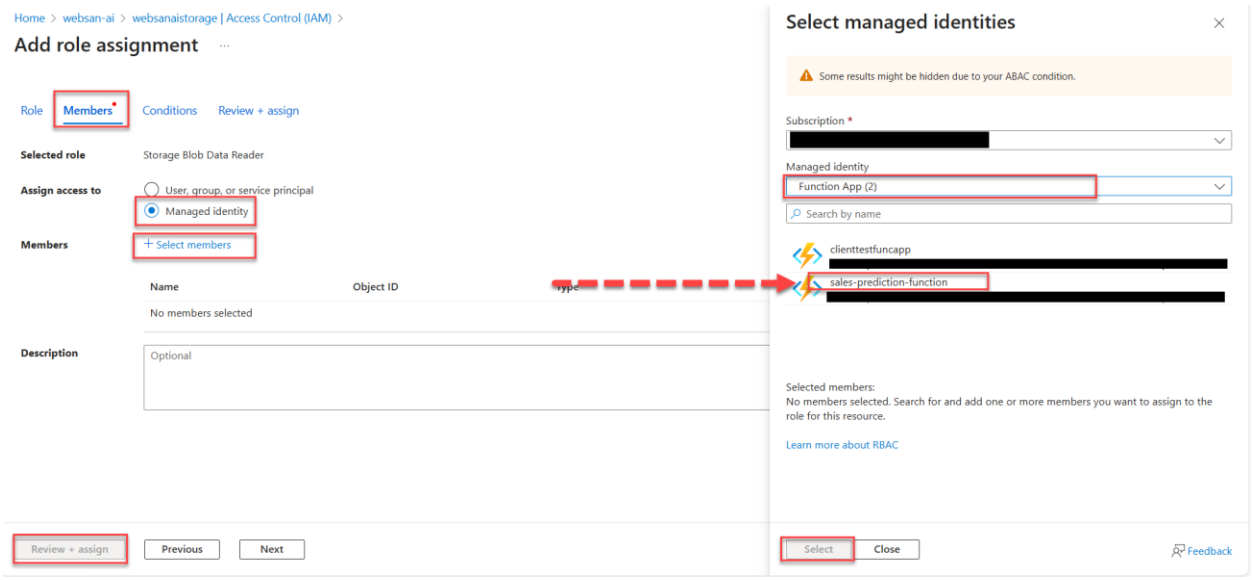
4. Go back to your Azure Data Lake Storage (ADLS) resource, on the left pane go to Access Control (IAM) > + Add > Add role assignment.



5. Select the Storage Blob Data **Contributor** role.



6. Under Members, select Managed identity, click +Select members, and select the Managed Identity for the Function App you created. Click Select, and Review + assign.

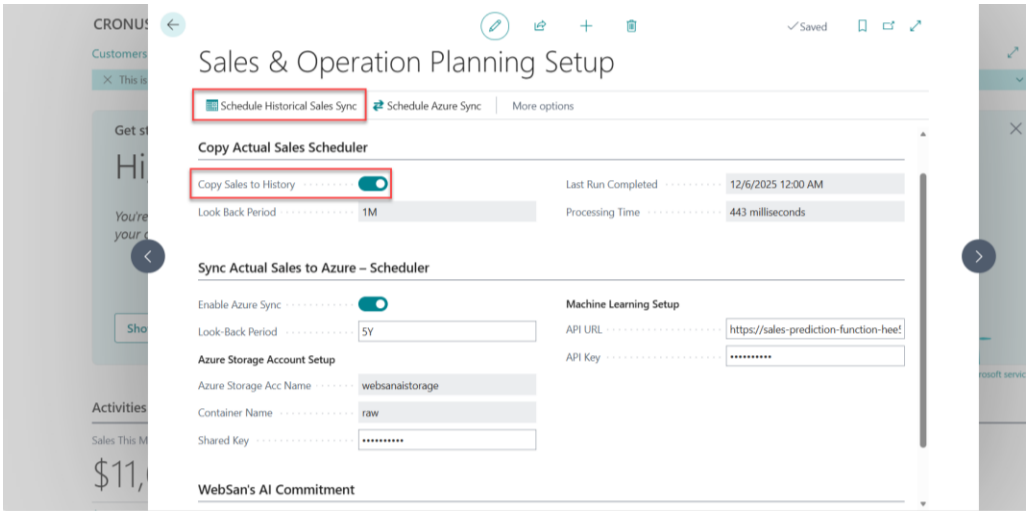


7. Go to the Business Central company where you will be using the AI Functionality of the Sales & Operations Planning application. Please note that the service is limited to a single company per installation.

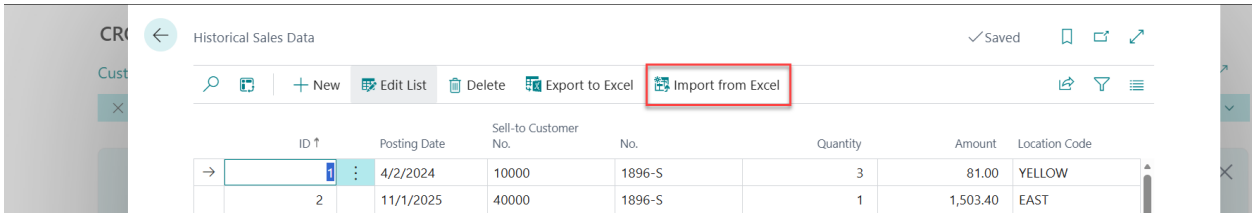
Business Central (i)

1. In Business Central, go to the Sales & Operations Planning Setup page. Activate the Copy Sales to History toggle and then use the Schedule Historical Sales Sync button at the top of the screen to schedule the job queue to run – you may also run it once on demand. This action will copy the data in the Posted Sales Invoice Lines to the Historical Sales Data page.

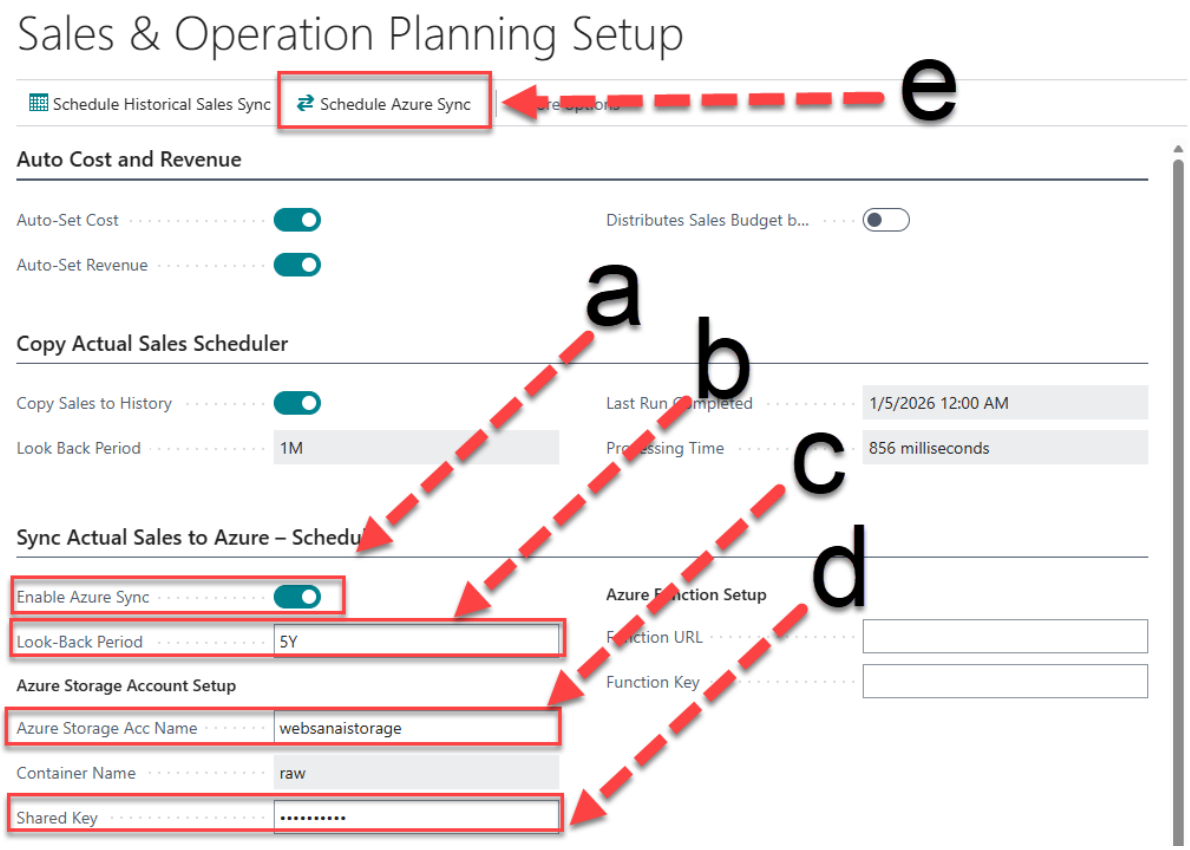
   The data loaded to the Historical Sales Data table will be used to train the AI forecasting model used by the application. Your data will never leave your Azure tenant or be used to train other clients' models.



2. Additionally, you may load historical data to your system by using the Import from Excel action found in the Historical Sales Data page. Download the Excel template (Export to Excel button), populate the file with your data, and reupload it using the Import from Excel action.

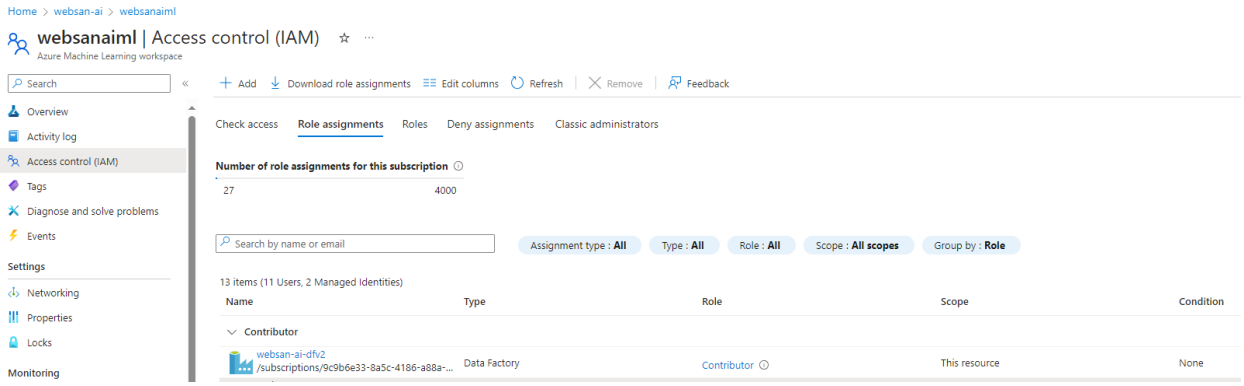


3. Go back to the Sales & Operations Setup page. Here you will need to:
   a. turn on the Enable Azure Sync toggle
   b. assign a value to the Look-Back Period in years (#Y)
   c. in the Azure Storage Acc Name field, enter the globally unique name you assigned to your Storage Account (ADLS)
   d. in the Shared Key field, enter the Storage Account key you temporarily saved to your computer
   e. click the Schedule Azure Sync button. This action will send the data in the Historical Sales Data table to the "raw" container in your Azure Data Lake Storage in Azure.



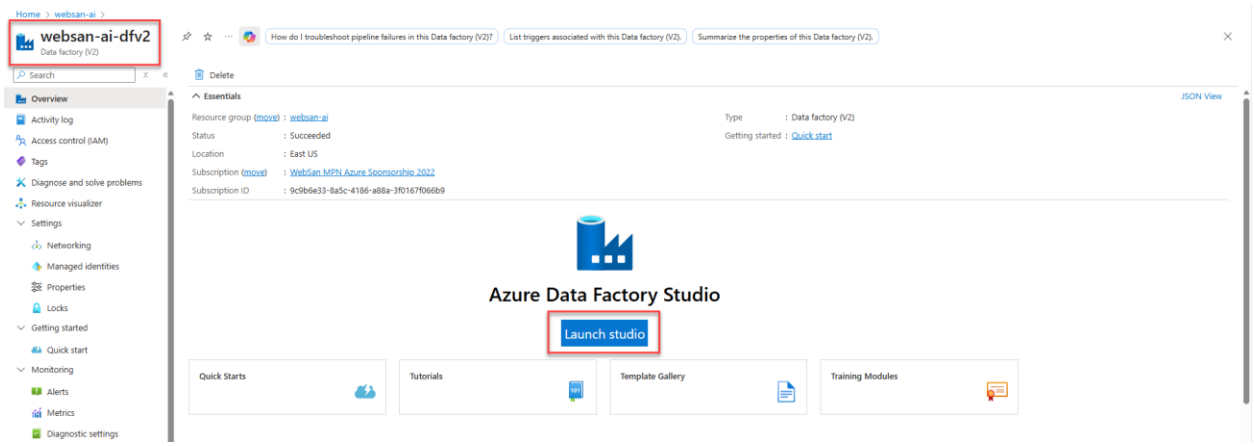
Azure Machine Learning Service (i)

1. In your Azure Portal, go to your websan-ai resource group and select your Azure Machine Learning Service resource.
2. Click the **Access Control (IAM)** menu in the sidebar, then add the Data Factory that you have created as a **Contributor**.

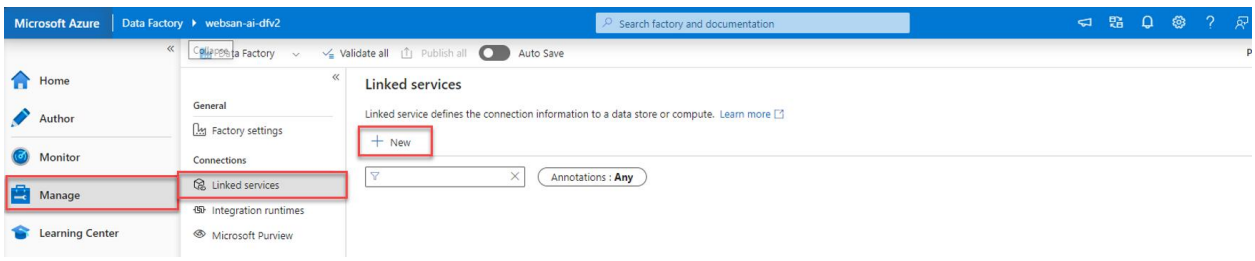


Azure Data Factory (i)

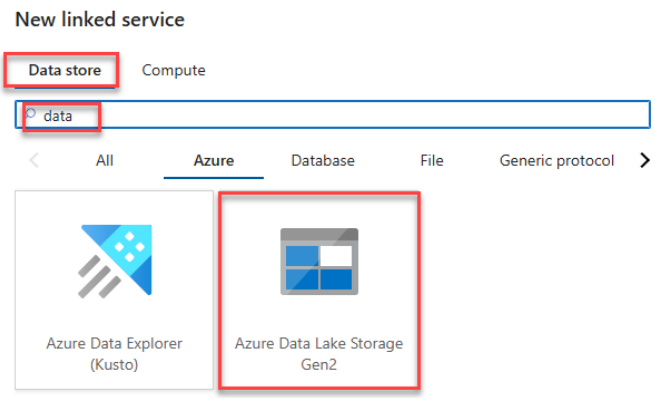
1. Go to your websan-ai resource group and select your Data Factory resource.
2. Click on Launch Studio. You will be redirected to the ADF studio.



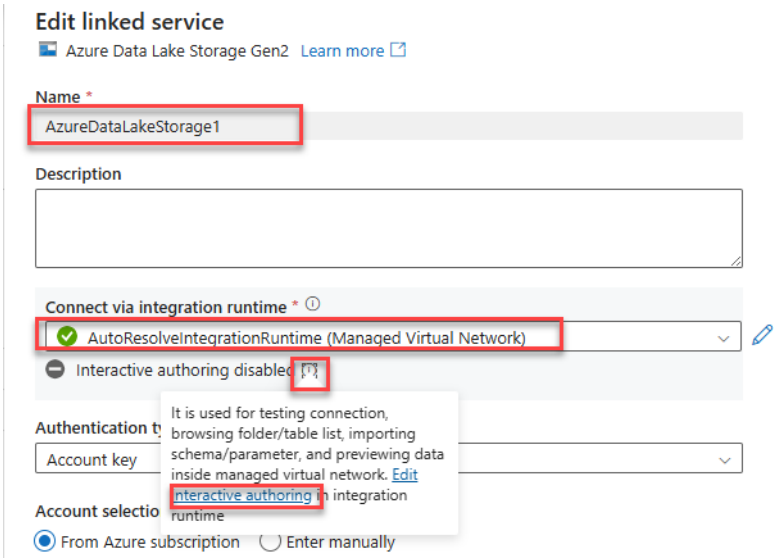
3. On the left-hand pane, click the Manage > Linked Services > + New. You will create two Linked services which will be used for setting the extract-load-transform (ETL) pipeline.



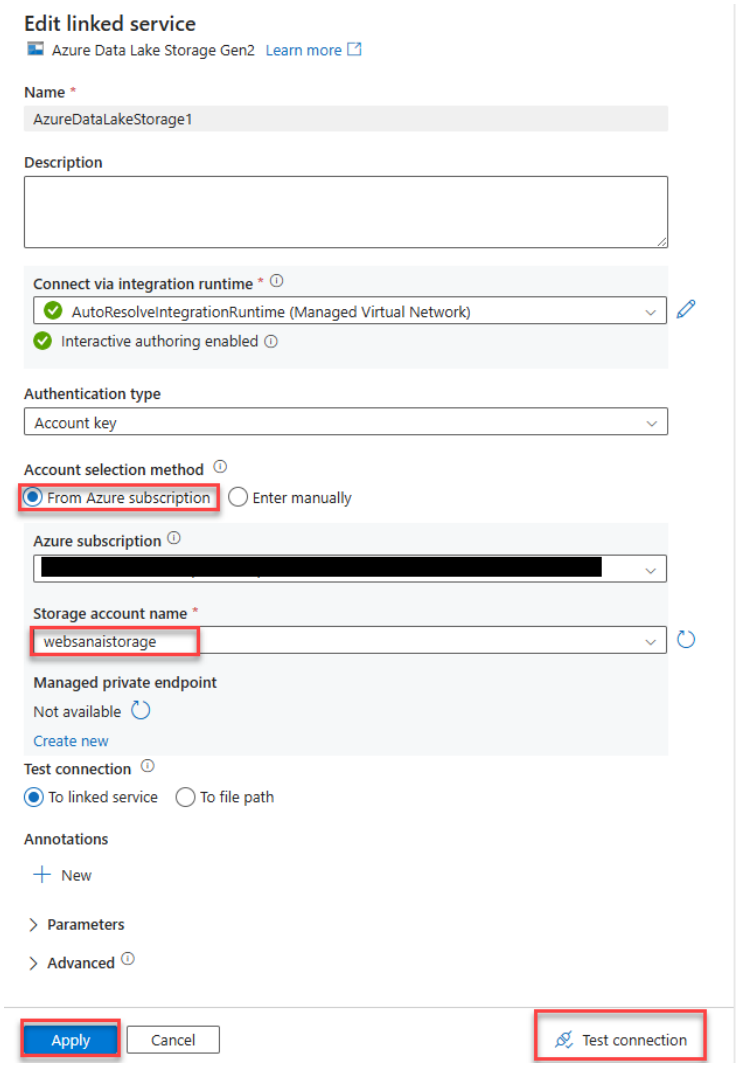
4. Search for the Azure Data Lake Storage Gen2 service in the Data Store tab, then click **Continue**.



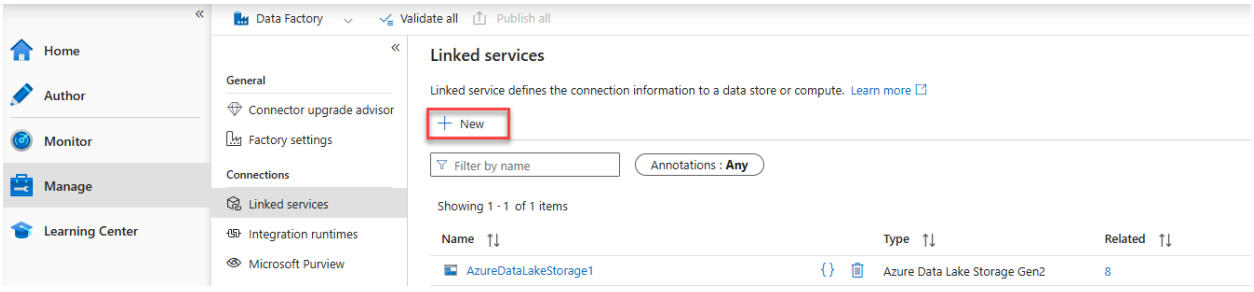
a. Configure the Linked Service.
    a. You may input your preferred name for the linked service
    b. Connect via integration runtime: AutoResolveIntegrationRuntime(Managed Virtual Network).
        i. Click the (i) button next to *Interactive authoring disabled*. Click the *Edit Interactive Authoring* link. In the window that appears, check the *Enable* box under Interactive authoring and hit Apply.



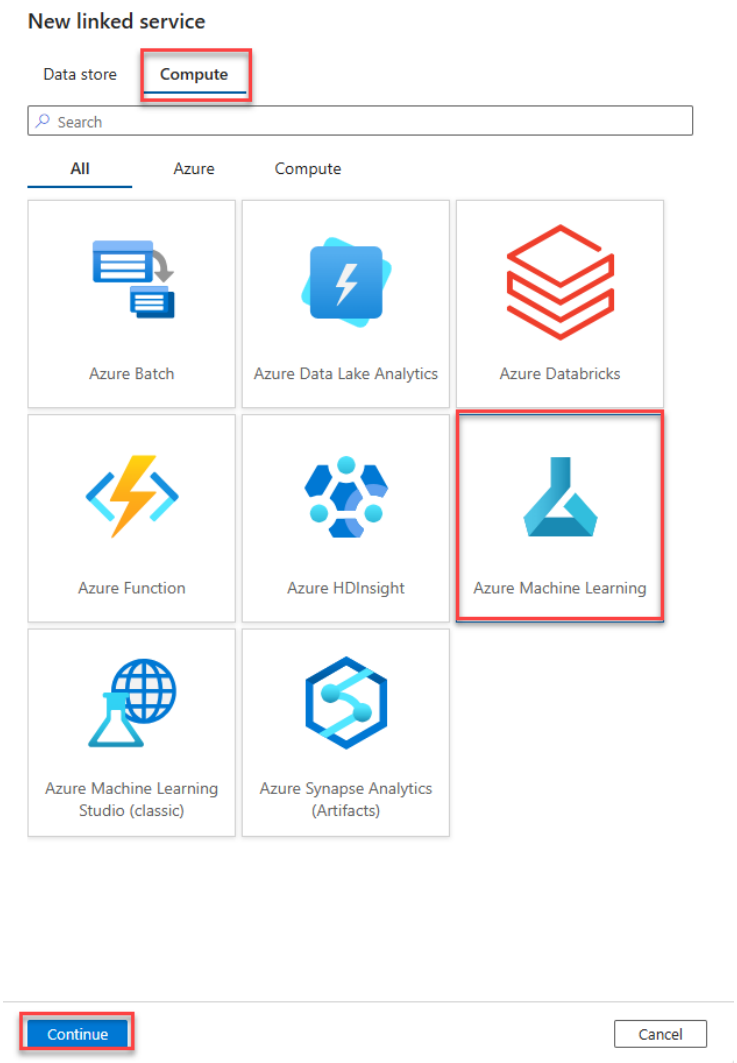
    c. Authentication type: Account key
    d. Account selection method: From Azure subscription
        i. Select your Azure subscription and Storage Account (ADLS) name.
    e. Click Test connection and then Apply.



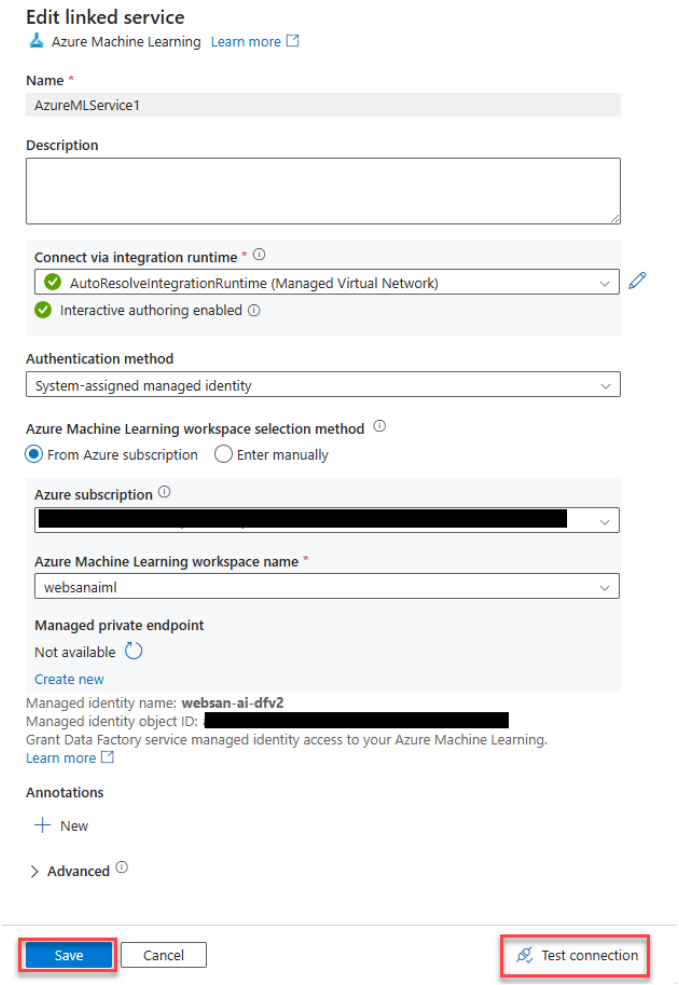
5. Click + New again.



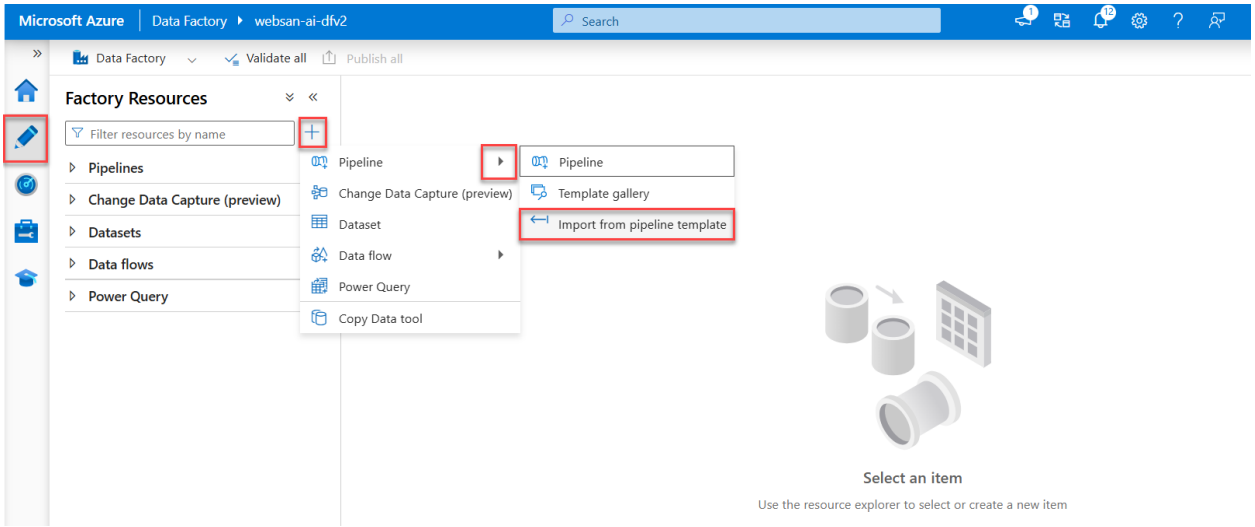
6. Search for the **Azure Machine Learning** in the Compute tab, then click **Continue**.



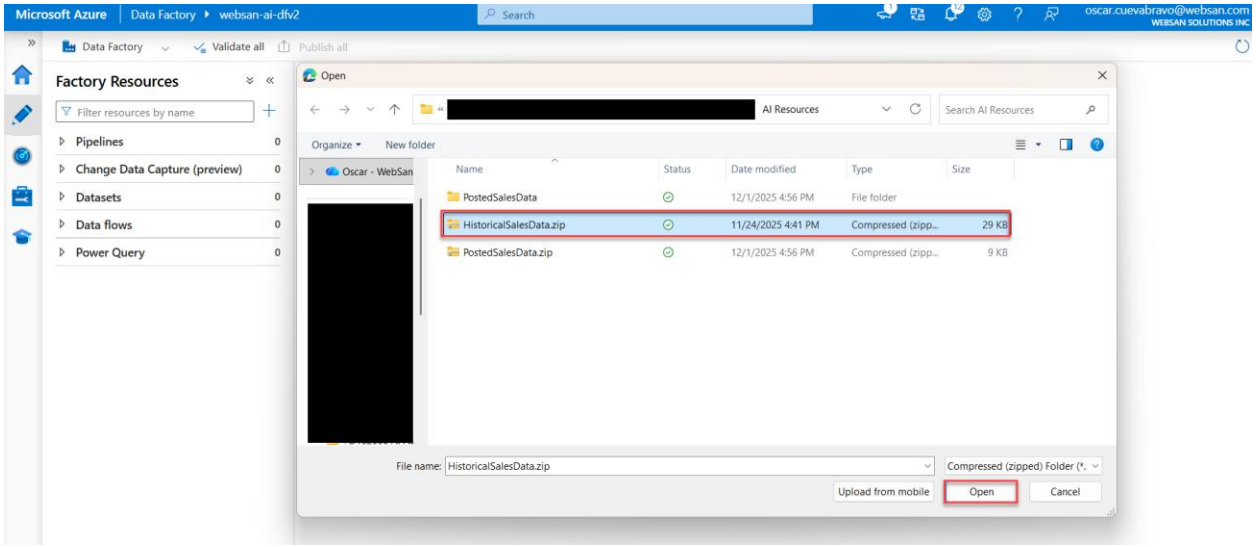
a. Configure the Linked Service.
    a. You may input your preferred name for the linked service
    b. Connect via integration runtime: AutoResolveIntegrationRuntime(Managed Virtual Network).
    c. Authentication type: System-assigned managed identity
    d. Account selection method: From Azure subscription
        i. Select your Azure subscription and Azure Machine Learning Workspace name (websanaiml).
    e. Click Test connection and then Save.



7. After setting up the Linked services, go to the Author menu, then click the plus button to add a pipeline. Hover to the Pipeline menu, then click the Import from pipeline template.

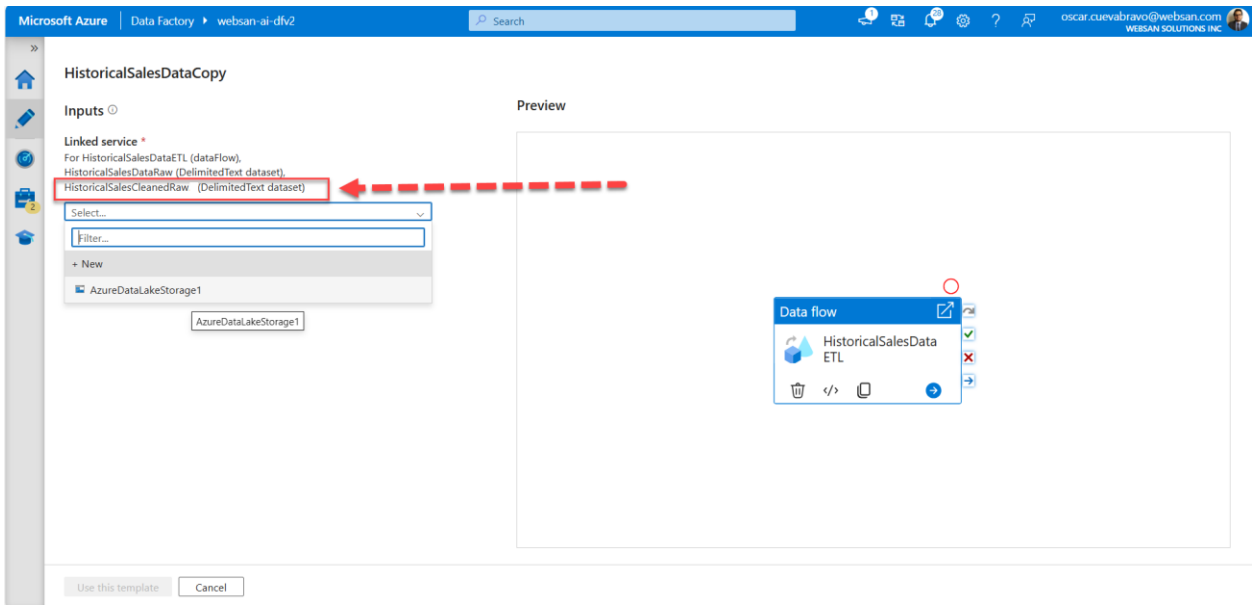


8. Use the *HistoricalSalesDataCopy.zip* **folder** as the template for the pipeline. This will create a pipeline, data flow, and datasets to be used in the pipeline. Select the linked services created in the inputs.

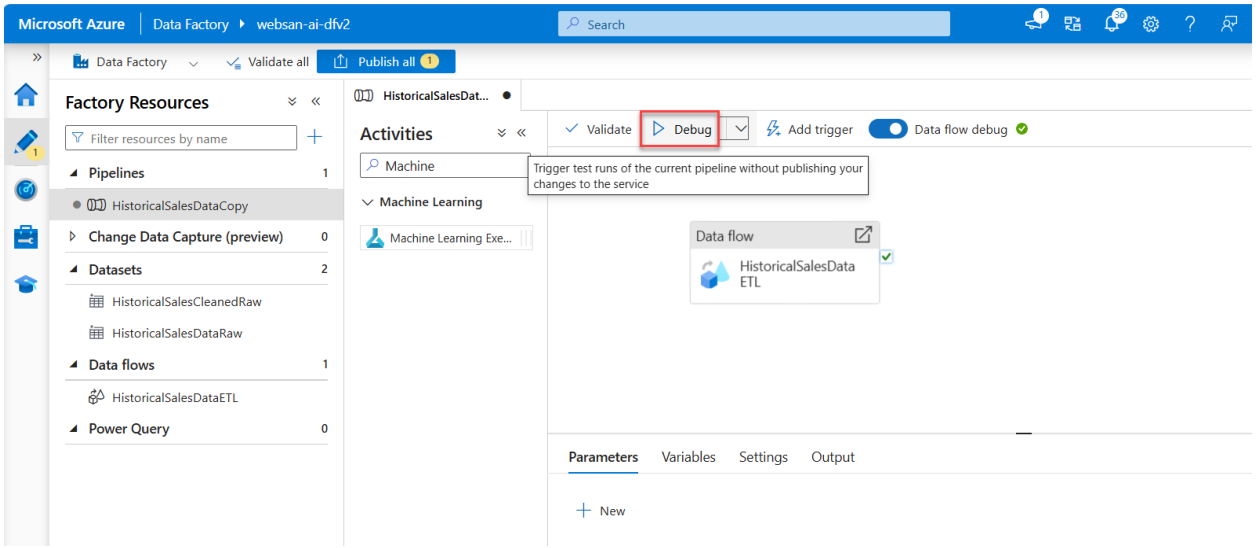


Remember: You must contact WebSan Support to obtain all installation files and scripts.

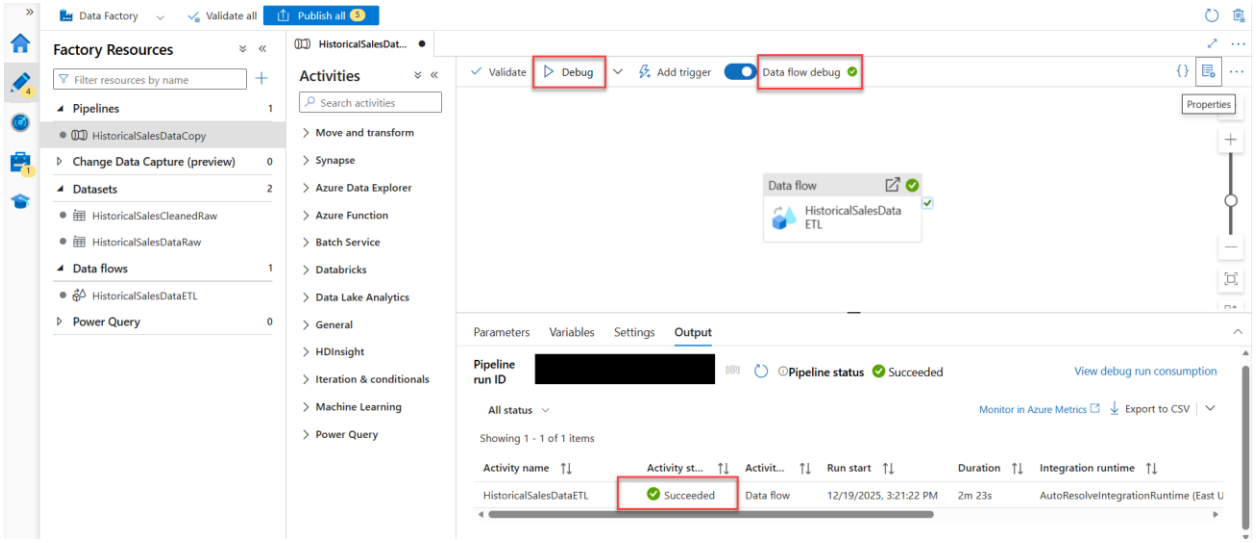
9. Select the Linked service and click Use this template.



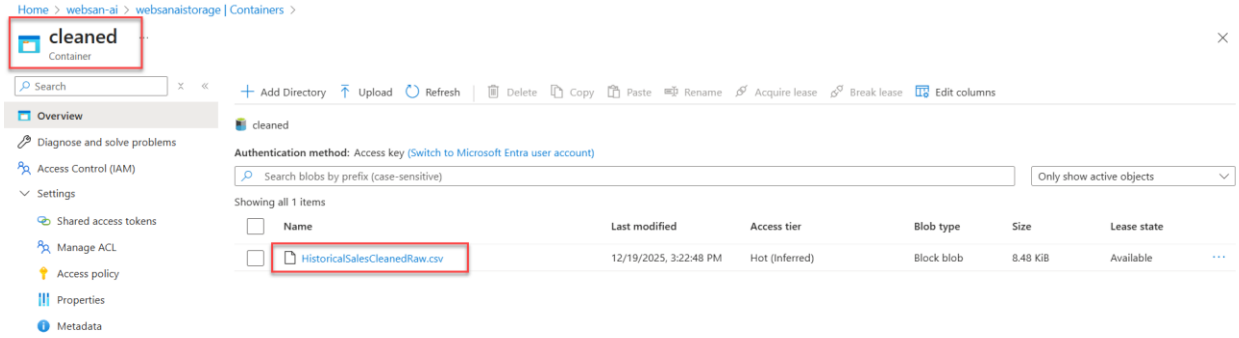
10. After creating the pipeline using the template, click the > Debug button to trigger the pipeline once without publishing. This will create the HistoricalSalesCleanedRaw.csv file in the Cleaned container in your Storage Account (ADLS), which the sales-data-forecast-model.py will use for training.



11. Wait until you see a green check mark next to Data flow debug and click the Debug button again. After a couple of minutes, you will see the status of the activity change to Succeeded.

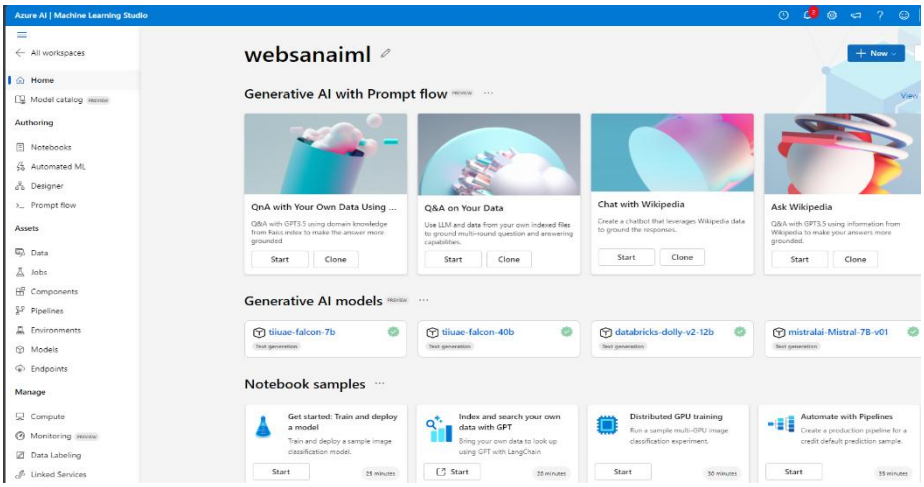


12. Optionally, can confirm that the HistoricalSalesCleanedRaw.csv file has been created by going to the Cleaned container in your ADLS resource.



Azure Machine Learning Service(ii)

1. Locate your AzureML resource and click the Launch Studio button. It will redirect you to a new window leading to the Azure Machine Learning workspace.



2. Under the Manage menu group, click the **Compute** menu. From here, we need to create a Compute instance.
   For user reference, this documentation was updated in December 2025. At the time of writing:
   - a *Standard_DS3_v2* compute *instance* was used for development and internal testing purposes.
   - a *STANDARD_D2as_V4* compute *cluster* named "WebsanMLCompute" is programmatically created when running the *pipeline-creator.ipynb* file. If the number of nodes in the cluster is not sufficient due to Microsoft updates; either:
     - comment out the cell block that creates the compute target (4th block) and create the compute manually in the Azure Machine Learning studio using the exact name indicated in the point above (*WebsanMLCompute*)
     - update the *vm_size* parameter in the block to allow for adequate number of nodes.

```
#Create or find the compute cluster used to run training jobs
compute_name = "WebsanMLCompute"
vm_size = "STANDARD_D2AS_V4"
```

```
(vm_size=vm_size,
 min_nodes=0, # Cluster can scale down to zero when idle
 max_nodes=4) # Maximum number of nodes allowed
```

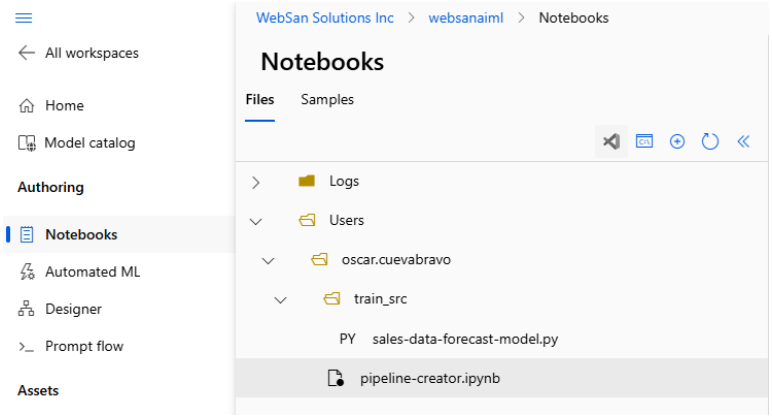
The compute instance and compute cluster listed above were used for development and testing purposes only. For large, production datasets, you *may* use more powerful compute resources. Note, however, that these will accrue more cost.

3. Click on the Notebooks menu under the Authoring menu group. From there, create a `train_src` folder inside the service account's folder. Afterwards, upload the *sales-data-forecast-model.py* file in the created folder.

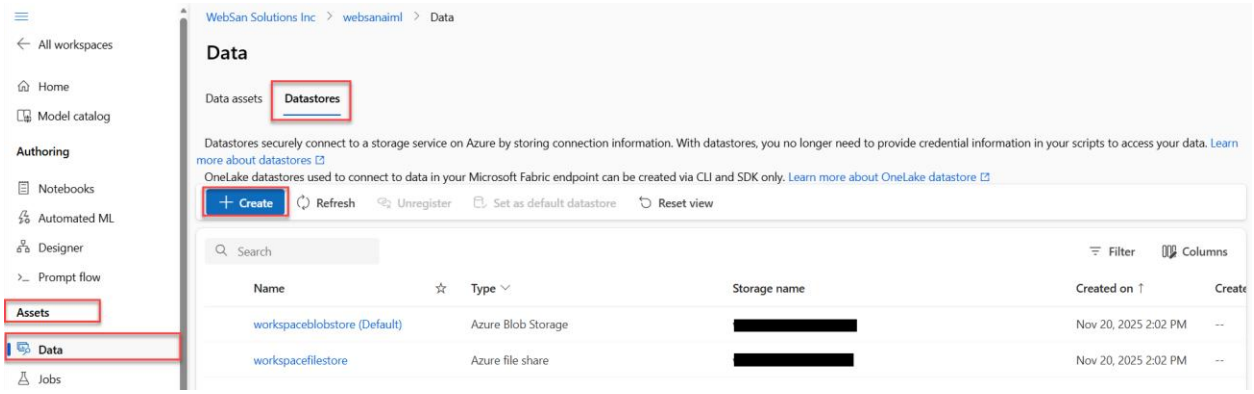
4. Upload the *pipeline-creator.ipynb* file to the root folder of the service account.

You must contact WebSan Support to obtain the installation files and scripts.

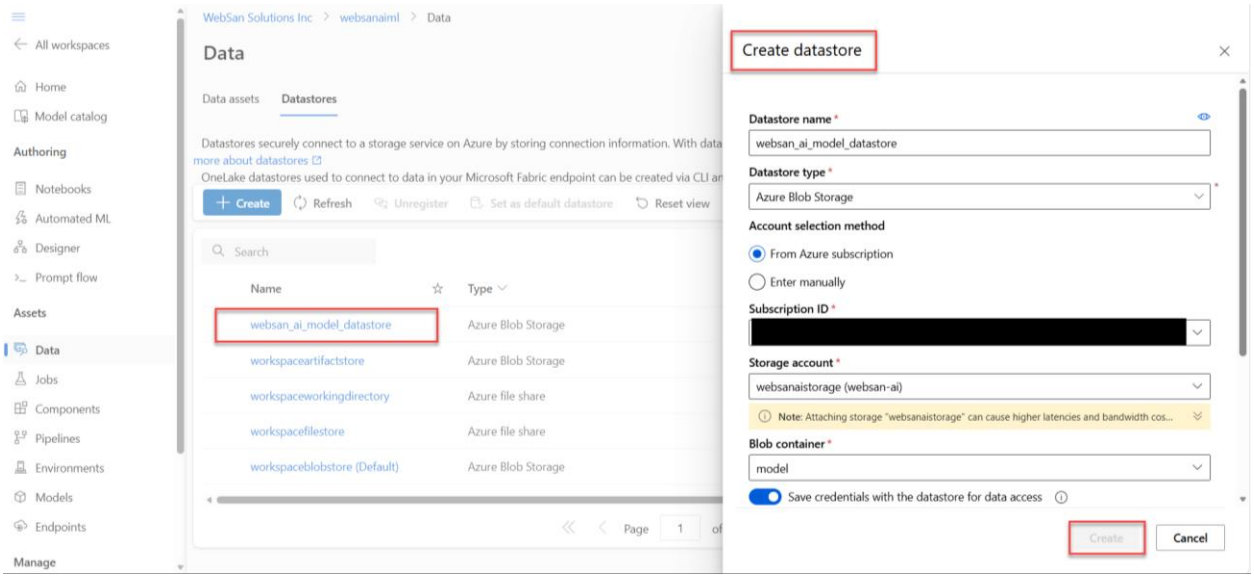
5. Confirm your folder output matches the below.



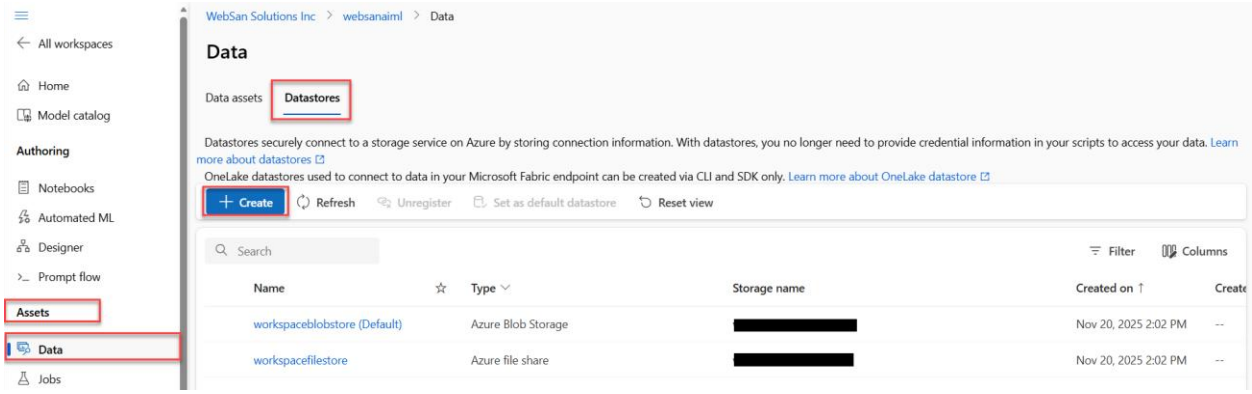
6. On the right-side pane, go to Assets > Data > Data Stores and click the + Create button.



   a. In the pane that appears, enter the following details:
      i. Datastore name: "*websan_ai_model_datastore*"
      ii. Datastore type: Azure Blob Storage
      iii. Account selection method: From Azure subscription
      iv. Select your Subscription ID
      v. Storage account: select your Storage Account
      vi. Blob container: select *model*
      vii. Authentication type: Account key
      viii. Account key: enter the key retrieved in the Azure Data Lake Gen2 step 1.
   b. Click Create
   c. Confirm the Datastore is listed

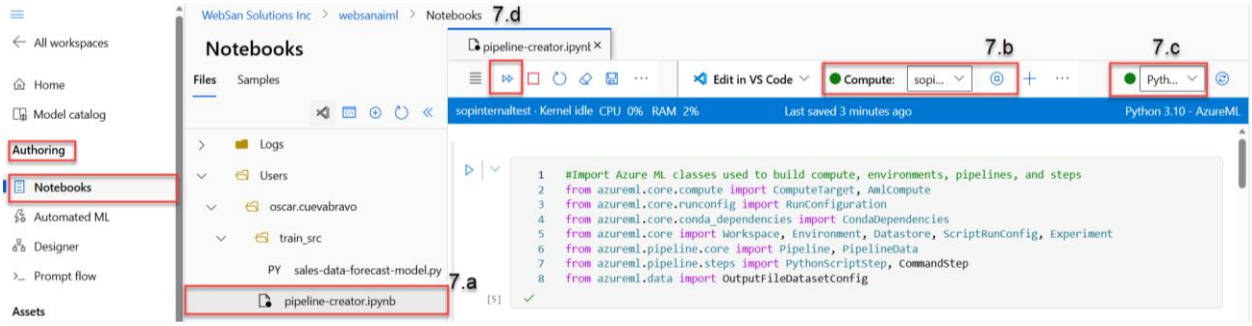


7. Click the + Create button again.



   a. In the pane that appears, enter the following details:
      i. Datastore name: "*websan_ai_cleaned_datastore*"
      ii. Datastore type: Azure Blob Storage
      iii. Account selection method: From Azure subscription
      iv. Select your Subscription ID
      v. Storage account: select your Storage Account
      vi. Blob container: select *cleaned*
      vii. Authentication type: Account key
      viii. Account key: enter the key retrieved in the Azure Data Lake Gen2 step 1.
   b. Click Create
   c. Confirm the Datastore is listed
8. To run the pipeline-creator.ipynb file you will need to:
   a. In the left-side pane, go to Authoring > Notebooks. Then, from your file tree, select the pipeline-creator.ipynb file.
   b. Ensure you have selected the correct compute instance and click the Play button. Wait for the compute to start.
   c. Set the kernel to *Python 3.10 – Azure ML*.

d. Click the >> button to execute the notebook and wait for the **entire** notebook run to complete.

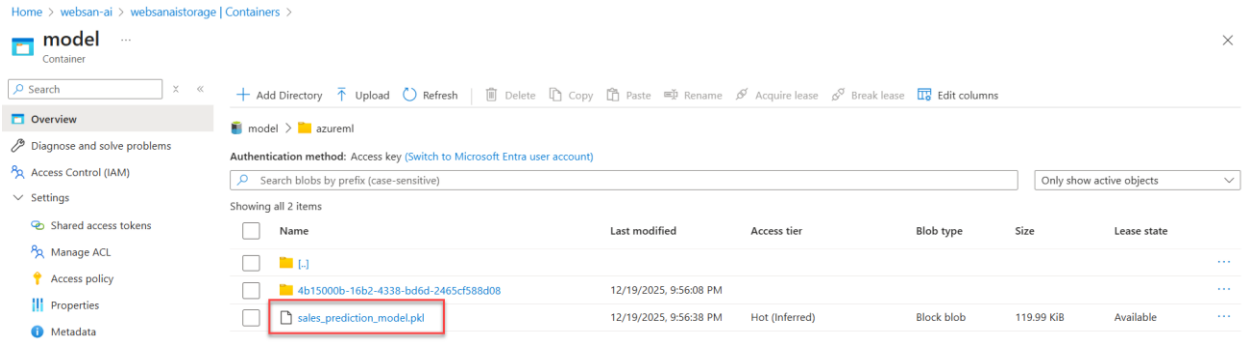


This notebook will create and/or access the following:

1. Compute cluster for the training
2. ML Pipeline for training
3. Trained model that is saved in the `model` folder in ADLS2 -> websan_ai_model_datastore in Azure Machine Learning studio allows the saving of the trained model artifact in the model folder in websanaistorage.

Expect slower training times on bigger datasets. Inform your AI engineer if you need faster training.

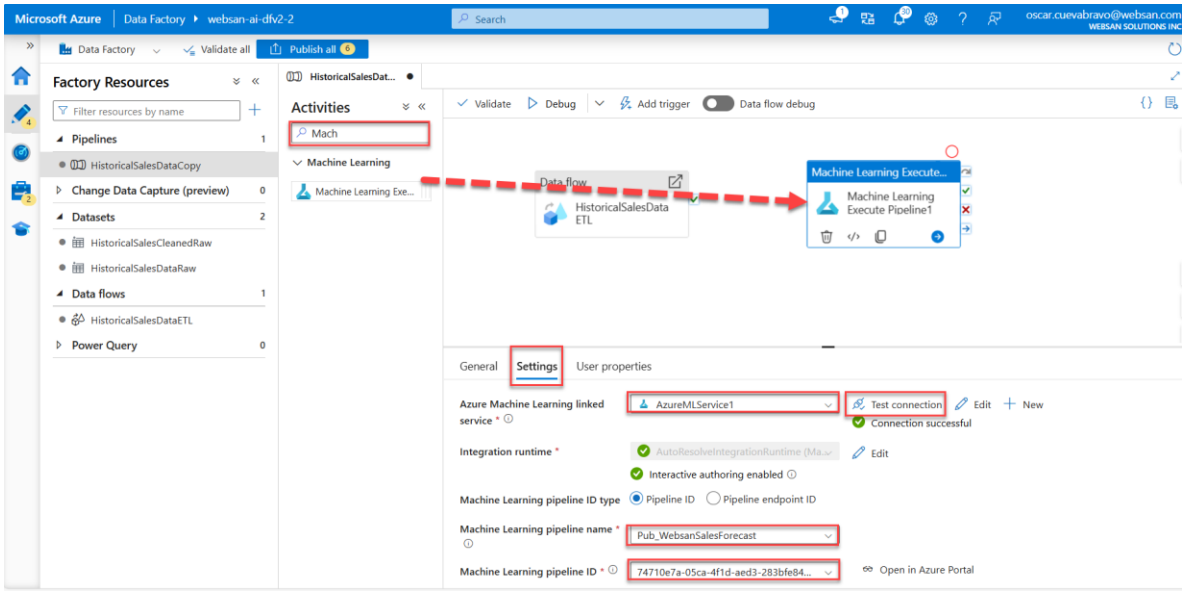
9. In Azure, go to your Data Storage (ADLS) resource. On the left-hand pane, go to Data Storage>Containers>model>azureml and select sales_prediction_model.pkl – confirm that the value in the Last Modified Date column corresponds to the date and time when training occurred.



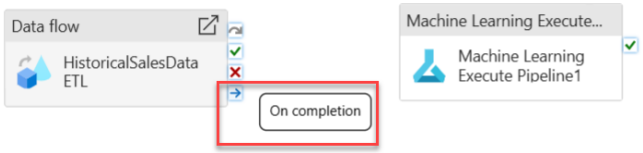
Azure Data Factory (ii)

1. To finalize the setup of your ADF, once your trained model (.pkl file) has been created, go back to your Azure Data Factory, open your pipeline and:
   a. Search for “machine learning execute” in the tab Activities, then drag the component into the work area.
   b. Double click the new component in the work area and go to the Settings tab
      i. For the Azure Machine Learning linked service, select the linked service you created at the beginning of this section. Test your connection.
      ii. For the ML Pipeline ID type, select the Pipeline ID

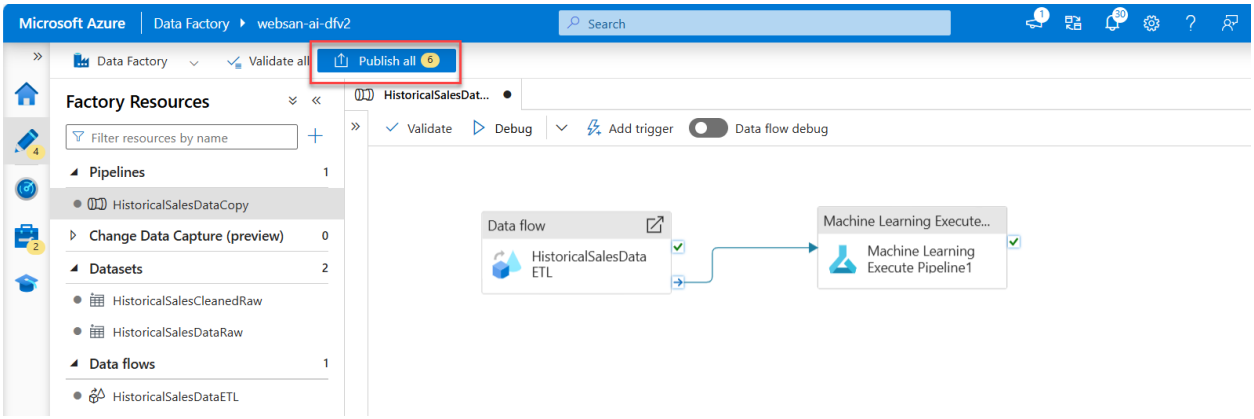
iii. Select the ML pipeline name that we created via Azure ML services.
iv. Select the ML pipeline ID that we created via Azure ML services.
v. (optional) Set the experiment name.



2. After setting the properties of the ML pipeline, connect the previous component to the ML component. Connection should be based on completion; use the green arrow as shown below, connect Data Flow to Machine Learning Execute.



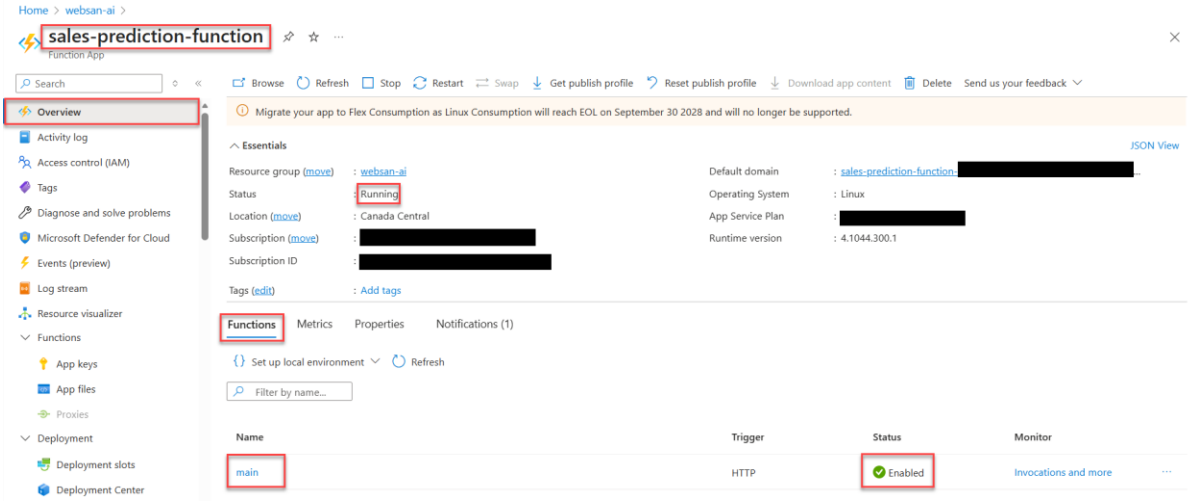
3. Final output should look like this:



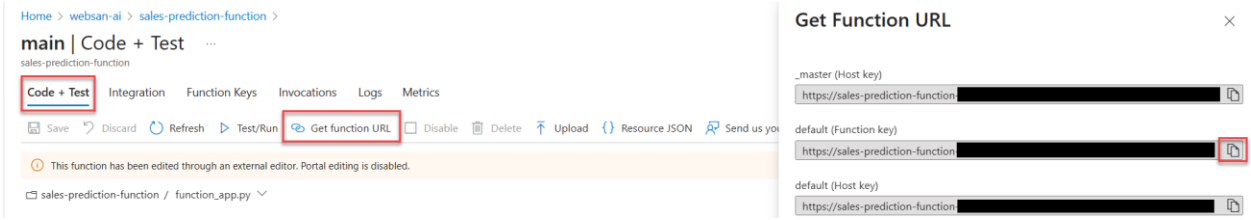
4. After adding the pipeline through the template and updating it with the new components, click the **Publish All** button at the topmost part of the workspace. For any validation errors, refer to the logs generated by the publisher.

Azure Function (ii)

1. You will need to contact the WebSan AI support team and provide them with the Azure login credentials to your tenant that you generated before. A Function will be published to your sales-prediction-function Function App resource.
2. You can confirm that the Function has been deployed by going to your Function app resource and locating the “main” function. Click on “main”.



3. Click on Get function URL. In the Get Function URL, click on the copy button for the **default (Function Key)**. Paste that key in a secure place only temporarily; you will use this key shortly.

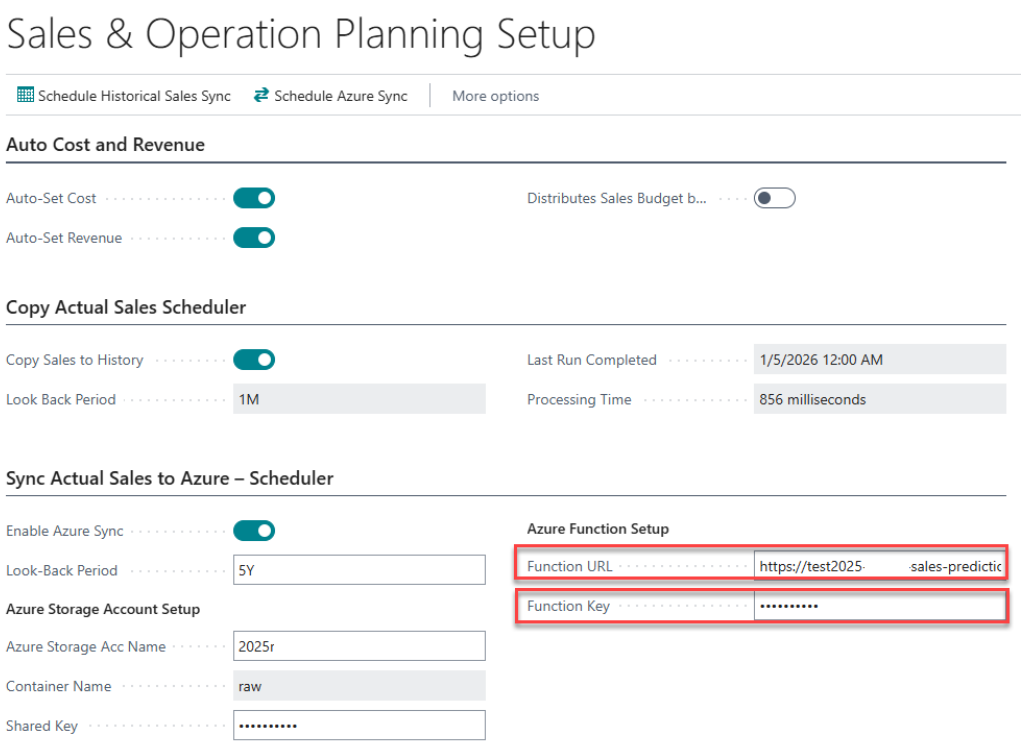


4. The Function Key will have the following format: https://<name_you_assigned_to_your_function>.azurewebsites.net/api/main~~?code=<key>~~

Business Central (ii)

To finalize the setup of the application, complete the following tasks:

1. In Business Central, go to the Sales & Operations Planning Setup page.
2. Take all the characters in the Function URL up to “main”, paste them in the Function URL field. ***https://<name_you_assigned_to_your_function>.azurewebsites.net/api/main***~~?code=<key>~~
3. Take all the characters in the Function Key after “code=”, paste them in the Function Key field. ~~https://<name_you_assigned_to_your_function>.azurewebsites.net/api/main?code=~~***<key>***

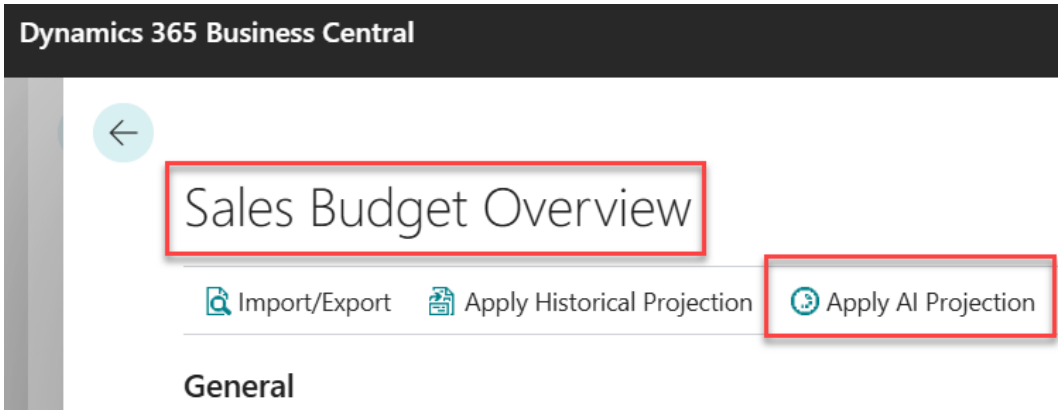


This concludes the setup of the AI resources for the Sales and Operations Planning application.

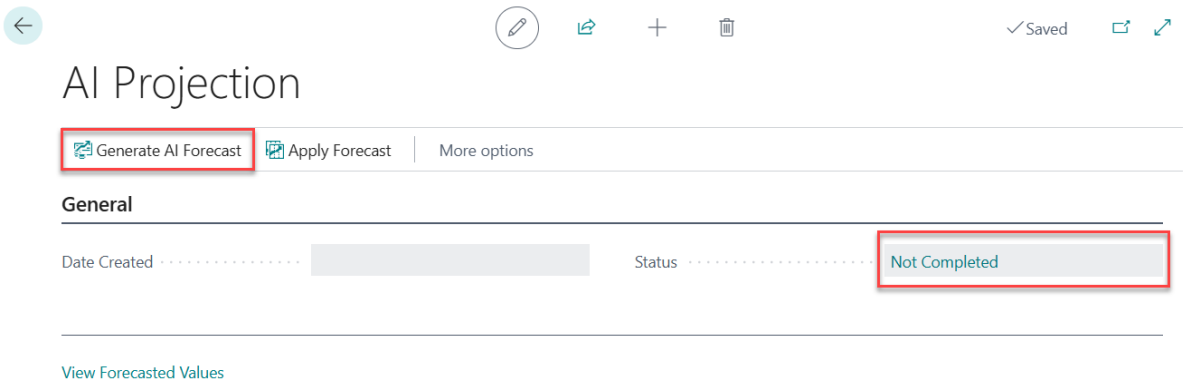
3. TESTING

Generation of First AI Forecast

In Business Central, go to Sales Budgets. Create a new budget or select one from the list. In the window that appears, select Apply AI Projection.



Click Generate AI Forecast. The Status field will be updated once the forecasting task is complete. You may close this page and come back to it later to check the status of the forecasting task.



4. MODEL RETRAINING

Retraining Fundamentals

Before you retrain your model, you must know the definition of two fundamental modelling concepts: forecast generation and model retraining.

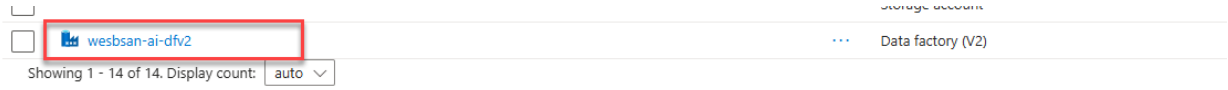
- **Forecast Generation**. To generate forecasts (or *predictions*), a model that clearly "understands" and accurately "represents" the underlying trends of the dataset used for training is required. While the market conditions (and as such, dataset underlying trends) remain the same as those that were present in the training data, it is best practice to continue using the same model.
- **Model retraining**. When current market conditions are "significantly" different to those present when the model was trained, it is important to start considering model retraining. Careful consideration to determine whether market conditions are different enough to merit the retraining of a model must always be made. While examples of these conditions are different for each business and vary greatly by industry, the following are common variables to track:
    - Interest rates
    - Taxes and/or tariffs
    - Competition
    - Technology changes
    - Labor market
    - Total/Fixed assets volume
    - Sustained decline in model output accuracy (actuals vs predicted)

You may refer to the following article published by IBM to learn more about model drift: https://www.ibm.com/think/topics/model-drift

Retraining Execution

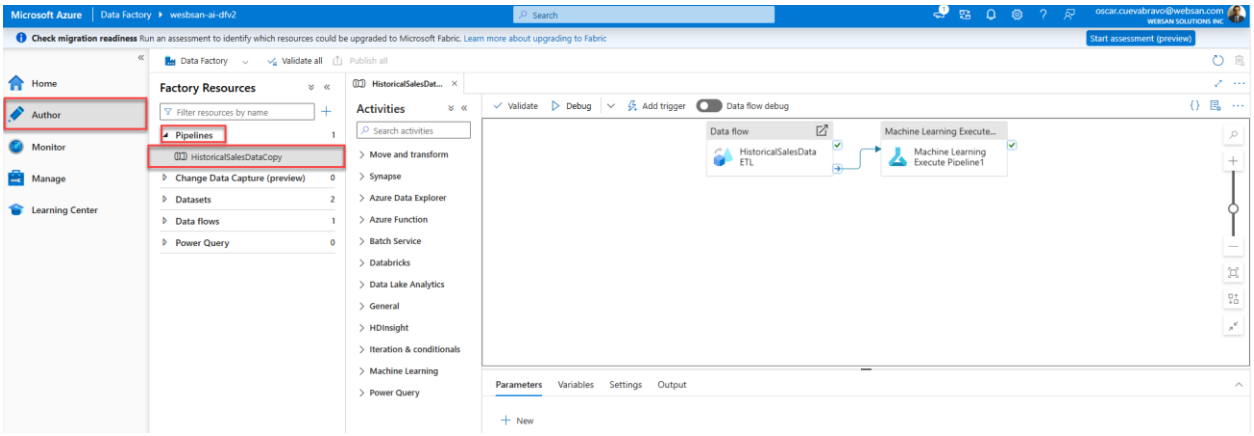
To retrain your production model, you will need to:

1. Make sure you have the latest version of your Business Central data available in your Azure tenant. You may refer to the following information to learn the steps required to sync Business Central data: Business Central (i)
2. Log in to your Azure tenant.
3. Go to your websan-ai resource group and locate your Azure Data Factory resource.

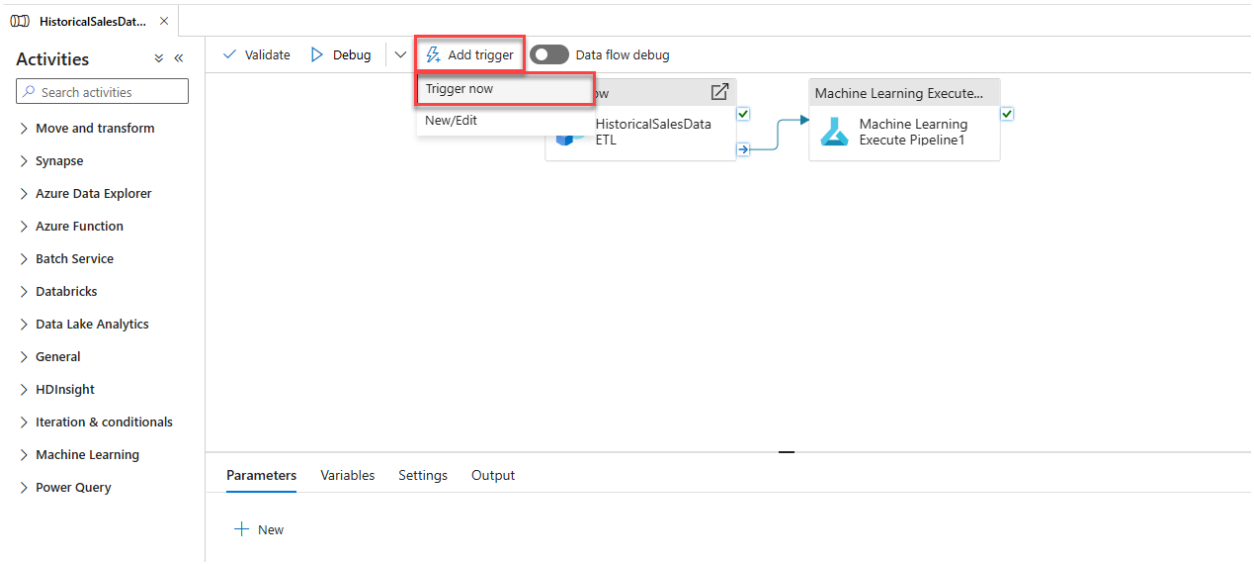


4. In the window that appears, click Launch Studio.

5. To open your pipeline, go to Author > Pipelines > HistoricalSalesDataCopy.



6. To trigger the pipeline, click on Add trigger > Trigger now.



7. This will pull the latest dataset found in the raw container, prepare it for model training, start a model training job in Machine Learning studio. A new sales_prediction_model.pkl file will be available.

A new model version will be available to generate AI sales forecasts generated from Business Central.