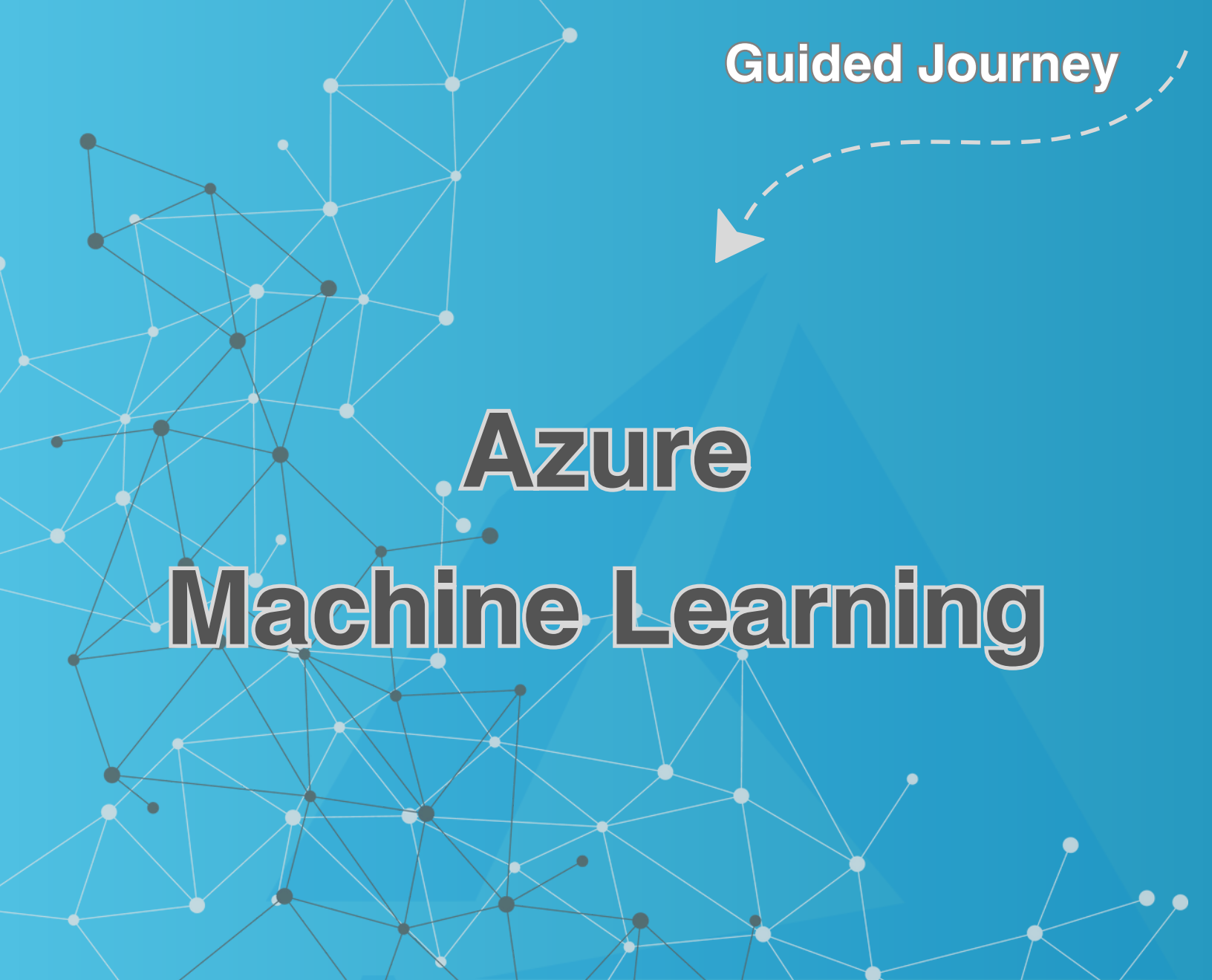


**Guided Journey**



# **Azure Machine Learning**

**Training, Deploying  
and Integrating  
Models**

**Jonathan Scholtes**

**July 2024**





# Azure Machine Learning - Training, Deploying and Integrating Models

This guide provides a comprehensive walkthrough for using Azure Machine Learning to build a classification model based on risk data pulled from Azure Fabric. You will learn how to preprocess the data, train the model, and register it within Azure Machine Learning. Additionally, the guide covers deploying the trained model as an online endpoint to enable real-time inferencing and incorporating it into web applications for seamless user interactions

- 
- ▶ [Architecture](#)
  - ▶ [Set-up](#)
  - ▶ [Model Training](#)
  - ▶ [Model Deployment and Integration](#)
- 



**Jonathan Scholtes**

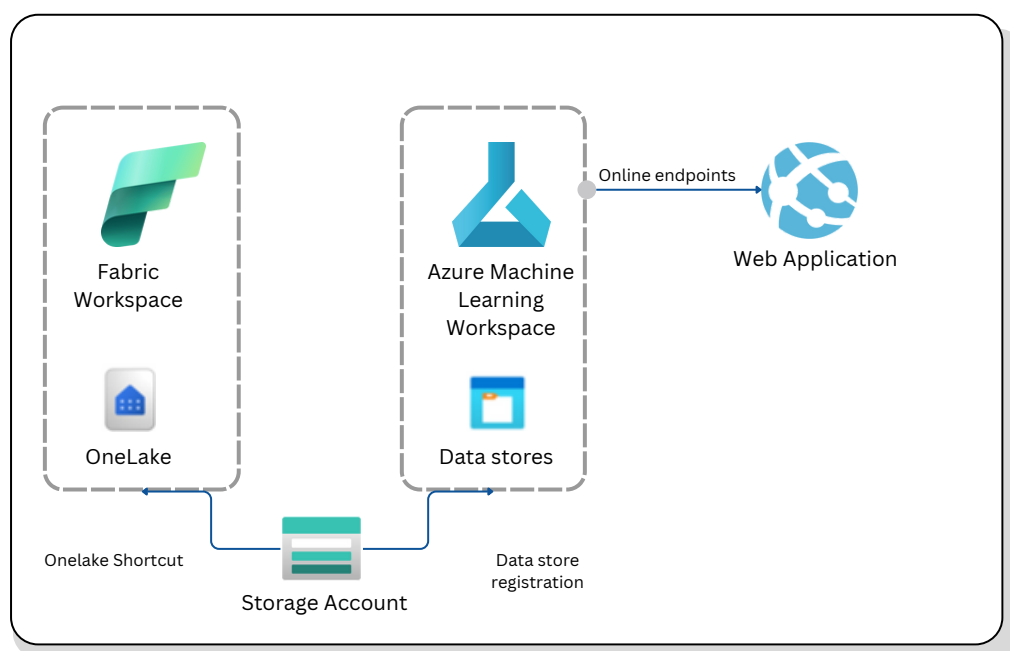
Cloud Solution Architect





# Architecture for Model Training and Integration

This architecture outlines the process of pulling risk data from Microsoft Fabric using a OneLake shortcut as a registered Azure Machine Learning datastore, creating a Tabular data asset for AutoML classification model training, and integrating the trained model with a web application.



## Data Ingestion and Storage

### Microsoft Fabric and OneLake

- Risk data is stored in Microsoft Fabric and accessed via a OneLake shortcut.
- The OneLake shortcut acts as a bridge, allowing Azure Machine Learning to directly interact with the data stored in Fabric

### Azure Machine Learning Datastore

- Register the OneLake shortcut as a datastore in Azure Machine Learning. This allows seamless access to the risk data for further processing.



## Data Preparation

### Tabular Dataset Creation

- Use the registered datastore to create a Tabular Dataset within Azure Machine Learning Studio.
- This dataset will be structured into rows and columns, suitable for machine learning and enable the use of AutoML.

## Model Training

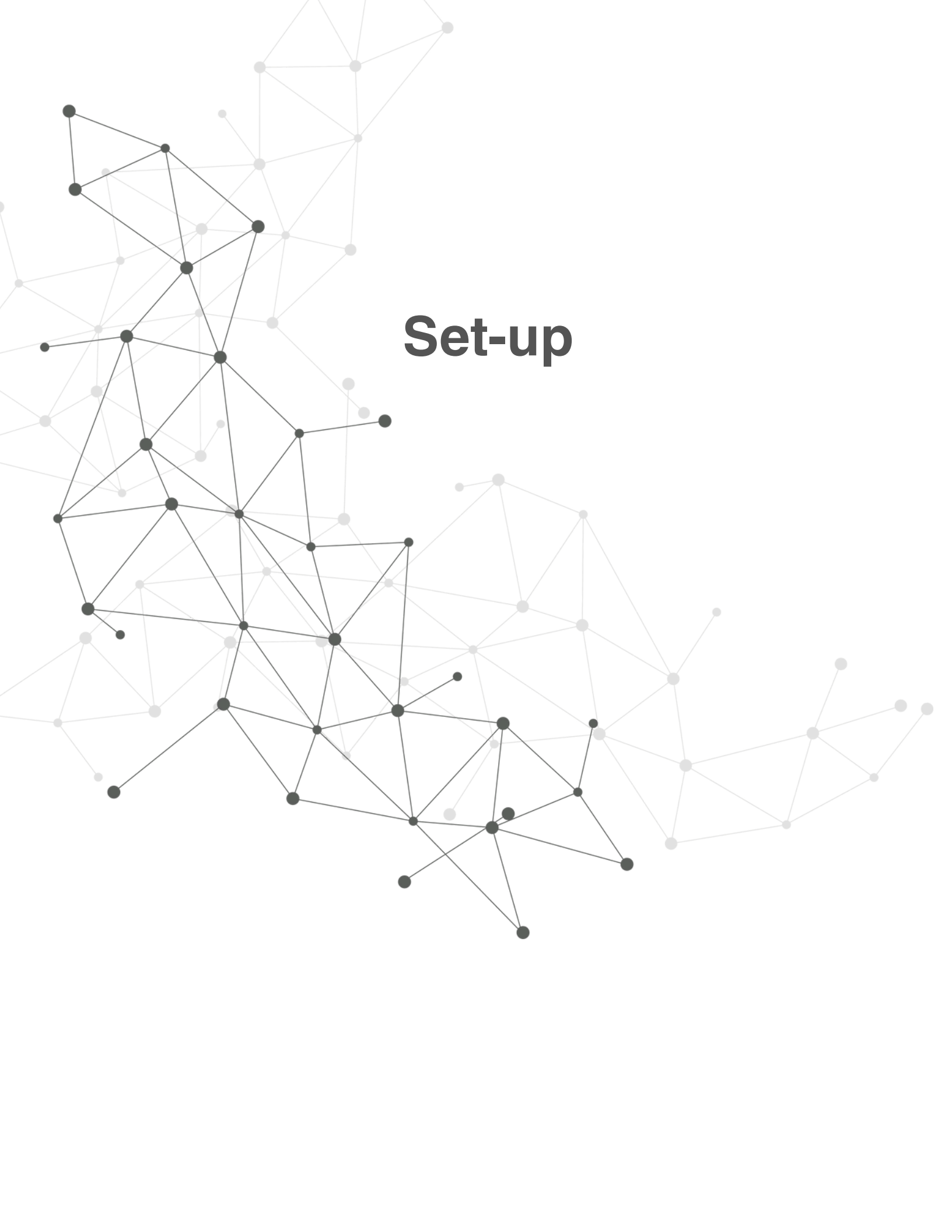
### AutoML Classification

- Use Azure Machine Learning's AutoML capabilities to train a classification model on the Tabular Dataset.
- AutoML will automate the selection, training, and tuning of the best-performing model for risk classification.

## Model Deployment and Integration

### Deployment to Online Endpoint

- Deploy the trained model to an Azure Machine Learning real-time online endpoint.
- This endpoint enables API access, allowing integration with existing web applications.
- Utilizing an online endpoint offers benefits such as limitless scaling, robust security with features like Private Link and customer-managed keys, and effective management through quota control, billing integration, and SLAs.

A network graph visualization on a white background. It consists of numerous nodes (dots) and edges (lines). The nodes are divided into two groups: dark grey/black nodes and light grey nodes. The dark nodes are concentrated in a dense, interconnected cluster on the left side of the image. From this cluster, several edges extend outwards to a more sparse network of light grey nodes that occupies the right side of the image. The overall structure suggests a transition or a relationship between a dense, established network and a more open, emerging one.

# Set-up

# Azure Machine Learning



Azure Machine Learning is a cloud service designed to accelerate and manage the entire machine learning (ML) project lifecycle. It is an invaluable tool for ML professionals, data scientists, and engineers, facilitating their daily workflows in training and deploying models, as well as managing machine learning operations (MLOps)

## Set up

Create a new Azure resource and search for **Azure Machine Learning**.



Please provide the **Resource Group**, **Name**, and **Region** for your Azure Machine Learning workspace.

**Azure Machine Learning** ...  
Create a machine learning workspace

Basics Networking Encryption Identity Tags Review + create

**Resource details**

Every workspace must be assigned to an Azure subscription, which is where billing happens. You use resource groups like folders to organize and manage resources, including the workspace you're about to create. [Learn more about Azure resource groups](#)

Subscription \* 

Resource group \*   [Create new](#)

**Workspace details**

Configure your basic workspace settings like its storage connection, authentication, container, and more. [Learn more](#)

Name \*   

Region \* 

Storage account \*   [Create new](#)

Key vault \*   [Create new](#)

Application insights \*   [Create new](#)

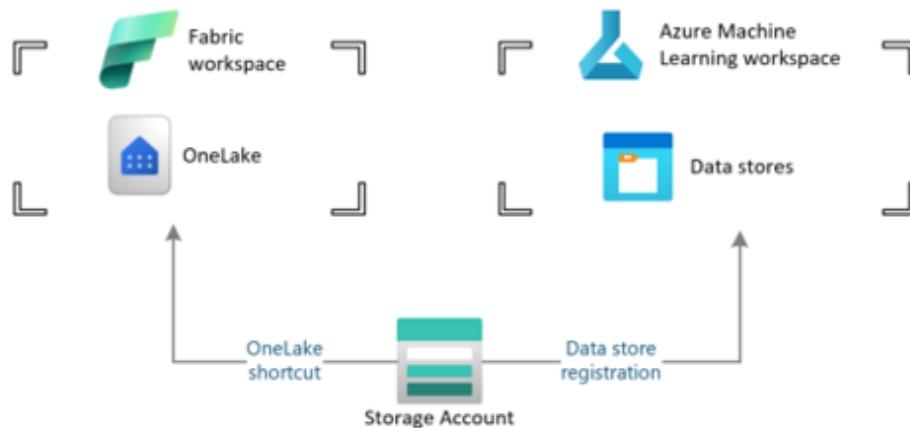
Container registry   [Create new](#)

The Azure Machine Learning Workspace requires an Azure Storage Account to store training run metrics and datasets, an Azure Container Registry for deploying model images, as well as Azure Key Vault and Application Insights for security and monitoring purposes.

**Review + create**

Click **Review + create**; For the purpose of this tutorial, we will use the default Networking, encryption and Identity setting

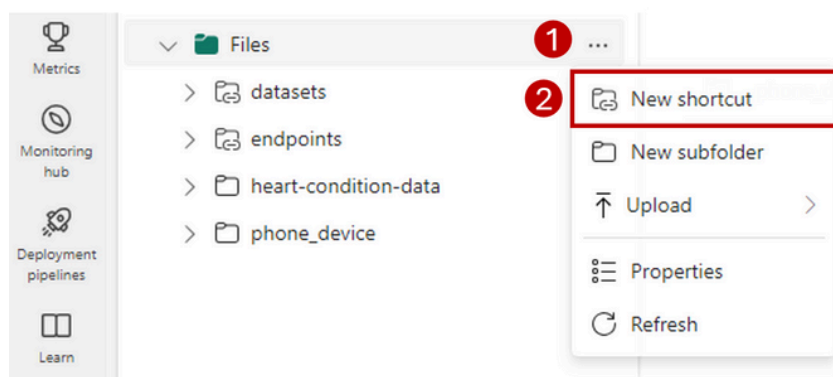
# Configure Fabric Data Access



To allow Fabric and Azure Machine Learning to read and write the same data without having to copy it, you can take advantage of OneLake shortcuts and Azure Machine Learning datastores. By pointing a OneLake shortcut and a datastore to the same storage account, you can ensure that both Fabric and Azure Machine Learning read from and write to the same underlying data.

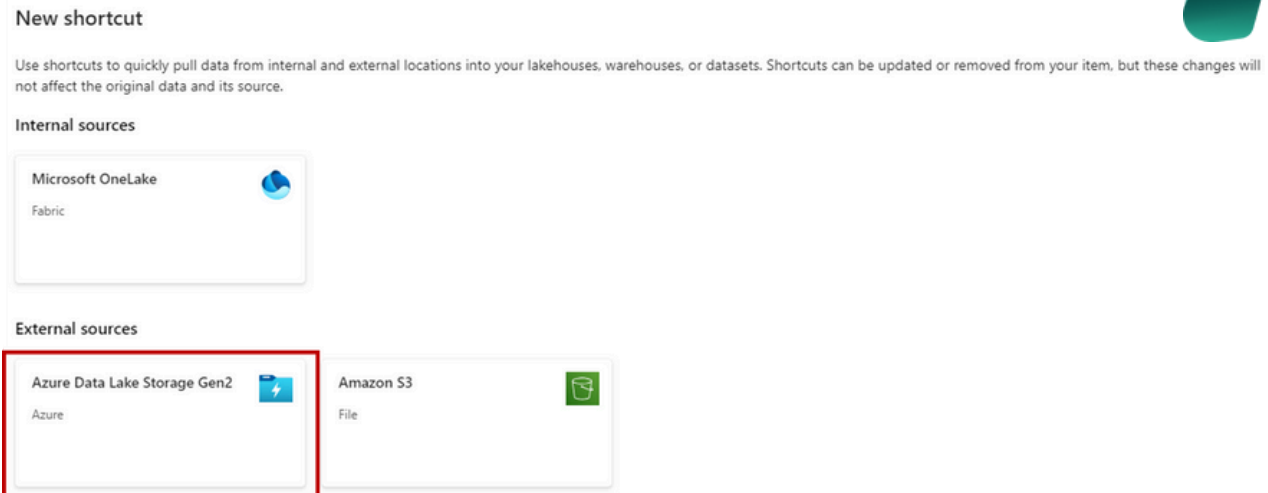
## Create a OneLake shortcut to the storage account

1. Open the **Synapse Data Engineering** experience in Fabric.
2. From the left-side panel, select your Fabric workspace to open it.
3. Open the lakehouse that you'll use to configure the connection. If you don't have a lakehouse already, go to the **Data Engineering** experience to create a lakehouse. In this example, you use a lakehouse named **trusted**.
4. In the left-side navigation bar, open more options for **Files**, and then select **New shortcut** to bring up the wizard.

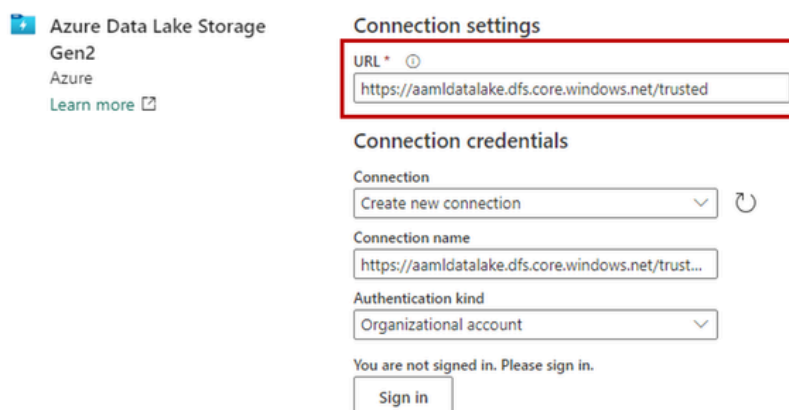




5. Select the **Azure Data Lake Storage Gen2** option.



6. In the **Connection** settings section, paste the URL associated with the Azure Data Lake Gen2 storage account.



7. In the **Connection credentials** section:

- For **Connection**, select **Create new connection**
- For **Connection name**, keep the default populated value.
- For **Authentication kind**, select **Organizational account** to use the credentials of the connected user via OAuth 2.0.
- Select **Sign in** to sign in.

8. Select **Next**.

9. Configure the path to the shortcut, relative to the storage account, if needed. Use this setting to configure the folder that the shortcut will point to.

10. Configure the **Name** of the shortcut. This name will be a path inside the lakehouse. In this example, name the shortcut **datasets**.

11. Save the changes.



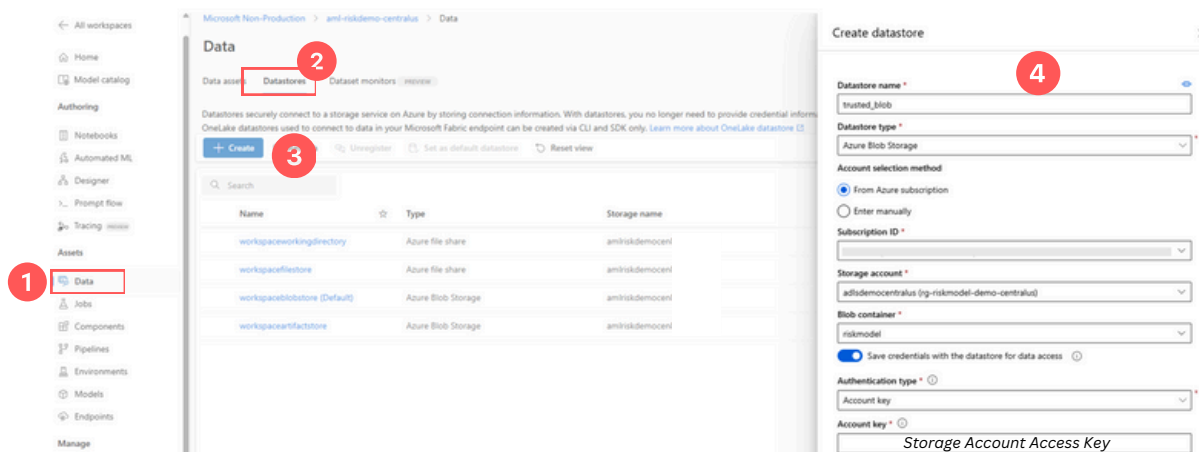
# Create Azure Machine Learning Datastore [🔗](#)

An Azure Machine Learning datastore acts as a reference to an existing Azure storage account, providing a common, easy-to-use API for interacting with various storage types (Blob, Files, ADLS). It simplifies the discovery of useful datastores for team operations and secures connection information for credential-based access (service principal, SAS, key), eliminating the need to include this information in your scripts.

## Create a datastore that points to the storage account

1. Go to your **Azure Machine Learning** workspace.
2. Go to the **Data** section.
3. Select the **Datastores** tab.
4. Select **Create**.
5. Configure the datastore as follows:
  - a. For Datastore name, enter **trusted\_blob**.
  - b. For Datastore type select **Azure Blob Storage**.
  - c. Select the storage account from the wizard, using the Subscription ID, Storage account, and Blob container (file system).

**Azure Data Lake Storage Gen2** can also be utilized for the Datastore type



Add permissions of a storage account to a Machine Learning workspace managed identity [🔗](#)

To preview datasets in Azure Machine Learning Studio using a storage account, you must enable "Use workspace managed identity for data preview and profiling" in the datastore settings.

Create

Click Create.



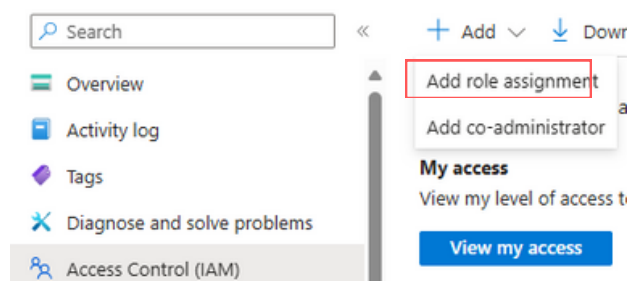
# Set-up Managed Identity Storage Account Access Control

Managed identity in Azure provides an automatically managed identity for services to authenticate securely with Azure AD without storing credentials.

The **Storage Blob Data Reader** role grants read-only access to Azure Blob storage data, allowing the managed identity to read blob content securely.

## Assign the Storage Blob Data Reader role to a managed identity

1. Navigate to the **Storage Account**
2. Access the **Access Control (IAM)** Section
3. Add a **Role Assignment**
  - a. Click on the "+ Add" button.
  - b. Choose "Add role assignment" from the dropdown



4. In the "Role" dropdown menu, type and select **Storage Blob Data Reader**.
5. Under "Assign access to," select "**Managed identity**."
6. Search for and select the managed identity linked to your **Azure Machine Learning** workspace.

### Add role assignment

1

Role **Members** Conditions Review + assign

Selected role Storage Blob Data Reader

Assign access to ☐ User, group, or service principal ☒ Managed identity 2

Members + Select members 3

| Name                   | Object ID | Type                             |
|------------------------|-----------|----------------------------------|
| aml-riskdemo-centralus |           | Azure Machine Learning workspace |

Review + assign

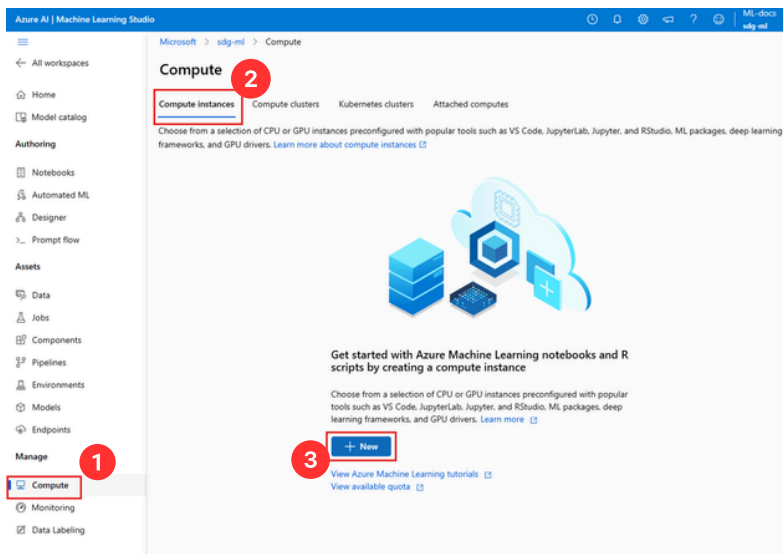
Click "Review + assign"



# Create Azure Machine Learning Compute Instance

Use a compute instance as a fully configured and managed development environment in the cloud. A compute instance can run multiple jobs in parallel and has a job queue. Since it's a dedicated development environment, it cannot be shared with other users in a workspace

1. Under **Manage**, select **Compute**.
2. Select **Compute instance** at the top.
3. If you have no compute instances, select **New** in the middle of the page



4. Configure **Required Settings**

| Field                | Description                                                                                                                                                                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Compute name         | Name is required (3-24 characters) and must start with a letter. Valid characters are upper/lower case letters, digits, and the '-' character, which must be followed by a letter. The name must be unique across all computes in the Azure region; you'll see an alert if it's not. |
| Virtual machine type | Choose CPU or GPU. This type can't be changed after creation                                                                                                                                                                                                                         |
| Virtual machine size | Supported virtual machine sizes might be restricted in your region. Check the availability list                                                                                                                                                                                      |



To avoid getting charged for a compute instance that is switched on but inactive, you can configure when to shut down your compute instance due to inactivity

- [enable idle shutdown](#)
- [add schedule](#)

**Review + Create**

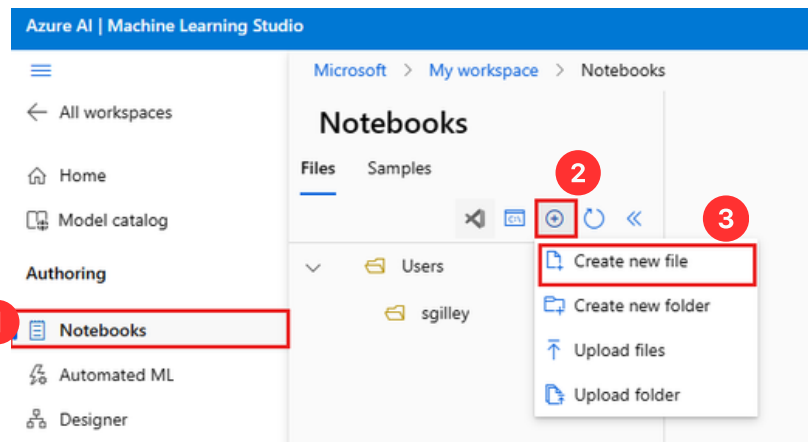
Click "Review + Create"



# Create Azure Machine Learning Notebook [🔗](#)

An Azure Machine Learning notebook is a web-based environment within Azure Machine Learning studio for interactive Python coding. It supports popular data science libraries and integrates seamlessly with Azure services, enabling data exploration, model training, and deployment directly in the Azure ecosystem

1. On the left side, select **Notebooks**.
2. Select the **+ tool**.
3. Select **Create new file**.



**Create new file**

File location  
Users/joschoit [Edit location](#)

File name \*

File type

☐ Overwrite if already exists

4. Name the file.
5. Select a file type - **Notebook (\*.ipynb)**
6. Select **Create**.

Select your recently created **Compute Instance** and a recent version of **Python**.

**Compute:**

**Python 3.10 - SDK v2**



# Create Azure Machine Learning Tabular Data Asset [🔗](#)

An Azure Machine Learning Tabular Data Asset (MLTable) organizes data into rows and columns, supporting sources like CSV files, SQL databases, and data lakes. This structured format is essential for AutoML, as it ensures clean data for automatic model selection, training, and tuning.

## Create handle to workspace [🔗](#)

Reference AML workspace with **MLClient**. Enter your Subscription ID, Resource Group name and Workspace name. To find these values:

1. In the upper right Azure Machine Learning studio toolbar, select your workspace name.
2. Copy the value for workspace, resource group and subscription ID into the code.
3. You'll need to copy one value, close the area and paste, then come back for the next one.



```
import time
import mltable
from mltable import MLTableHeaders, MLTableFileEncoding, DataType
from azure.ai.ml import MLClient
from azure.identity import DefaultAzureCredential
from azure.ai.ml.entities import Data
from azure.ai.ml.constants import AssetTypes

# authenticate
credential = DefaultAzureCredential()

ml_client = MLClient(credential=credential, subscription_id="<SUBSCRIPTION_ID>",
resource_group_name="<RESOURCE_GROUP>",
workspace_name="<AML_WORKSPACE_NAME>", )
```

## Access data from a datastore URI [🔗](#)

A Datastore URI is a Uniform Resource Identifier, which is a reference to a storage location (path) on your Azure storage account. You can copy-and-paste the datastore URI from the Studio UI with these steps

1. Select Data from the left-hand menu, then select the Datastores tab.
2. Select your datastore name, and then Browse.
3. Find the file/folder you want to read into Pandas, and select the ellipsis (...) next to it. Select Copy URI from the menu.

### Copy upload path URI



Choose a URI format to copy.

**Datastore URI**

Storage URI

The datastore URI references a data storage location within a datastore. This URI utilizes the credentials stored in the datastore, and can only be used within Azure Machine Learning. It is particularly useful as the input path property inside a job definition.

azureml://subscriptions/

**Copy**



```
# Path to your data within the Datastore - pasted from Datastore URI
data_path = [{"file": "azureml://subscriptions/<subscription_id>/resourcegroups/rg-
riskmodel-demo-centralus/workspaces/aml-riskdemo-
centralus/datastores/trusted_blob/paths/credit_risk_dataset.csv"}]
```



## Create MLTable and Data Asset

Create an MLTable and a data asset in Azure Machine Learning. Creating a data asset uploads the MLTable to cloud storage and bookmarks it, allowing team members to access it with a friendly name. Additionally, the data asset is versioned for easy tracking and management.



```
# create an MLTable from the data files
tbl = mltable.from_delimited_files(
    paths=data_path,
    delimiter=";",
    header=MLTableHeaders.all_files_same_headers,
    infer_column_types=True,
    include_path_column=False,
    encoding=MLTableFileEncoding.utf8,
)

# save the data loading steps in an MLTable file
mltable_folder = "./credit_risk"
tbl.save(mltable_folder)

# set the version number of the data asset to the current UTC time
VERSION = time.strftime("%Y.%m.%d.%H%M%S", time.gmtime())

# show the first few records
print(tbl.show())

# Define the Data asset object
my_data = Data(
    path=mltable_folder,
    type=AssetTypes.MLTABLE,
    description="credit risk table created from exiting store",
    name="credit_risk",
    version=VERSION,
)

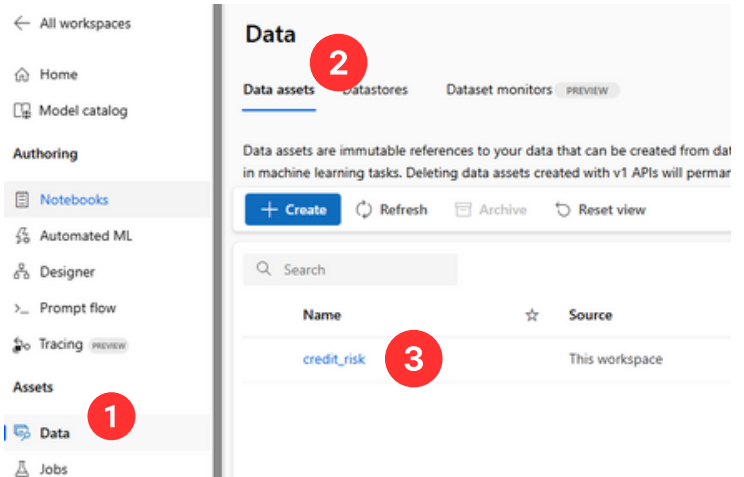
# Create the data asset in the workspace
ml_client.data.create_or_update(my_data)
```



# Explore Azure Machine Learning Tabular Data Asset

Now that we've created our Tabular Data Asset (MLTable), let's explore the data it contains:

1. Click on **Data** on the left sidebar.
2. Select **Data assets**.
3. Choose the specific data asset (e.g., `credit_risk`).
4. Click on **Explore** to delve into the data.



The screenshot shows the 'credit\_risk' data asset details page. The 'Explore' tab is selected with a red circle labeled '4'. Below the tab, there is a table of data. The table has 12 columns and 50 rows. The first 10 rows are shown in the screenshot.

| person_age | person_income | person_home_o... | person_emp_le... | loan_intent | loan_grade | loan_amnt | loan_int_rate | loan_status | loan_percent_in... | cb_person_defa... | cb_person_c |
|------------|---------------|------------------|------------------|-------------|------------|-----------|---------------|-------------|--------------------|-------------------|-------------|
| 22         | 59000         | RENT             | 123              | PERSONAL    | D          | 35000     | 16.02         | true        | 0.59               | true              | 3           |
| 21         | 9600          | OWN              | 5                | EDUCATION   | B          | 1000      | 11.14         | false       | 0.1                | false             | 2           |
| 25         | 9600          | MORTGAGE         | 1                | MEDICAL     | C          | 5500      | 12.87         | true        | 0.57               | false             | 3           |
| 23         | 65500         | RENT             | 4                | MEDICAL     | C          | 35000     | 15.23         | true        | 0.53               | false             | 2           |
| 24         | 54400         | RENT             | 8                | MEDICAL     | C          | 35000     | 14.27         | true        | 0.55               | true              | 4           |
| 21         | 9900          | OWN              | 2                | VENTURE     | A          | 2500      | 7.14          | true        | 0.25               | false             | 2           |
| 26         | 77100         | RENT             | 8                | EDUCATION   | B          | 35000     | 12.42         | true        | 0.45               | false             | 3           |
| 24         | 78956         | RENT             | 5                | MEDICAL     | B          | 35000     | 11.11         | true        | 0.44               | false             | 4           |
| 24         | 83000         | RENT             | 8                | PERSONAL    | A          | 35000     | 8.9           | true        | 0.42               | false             | 2           |
| 21         | 10000         | OWN              | 6                | VENTURE     | D          | 1600      | 14.74         | true        | 0.16               | false             | 3           |

An abstract network diagram featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, with a subset of them being black and the others being light gray. The lines connecting the nodes are thin and black, creating a dense, interconnected structure. The overall shape of the network is irregular and sprawling, with many branches and loops. The text "Model Training" is overlaid on the right side of the image.

# Model Training

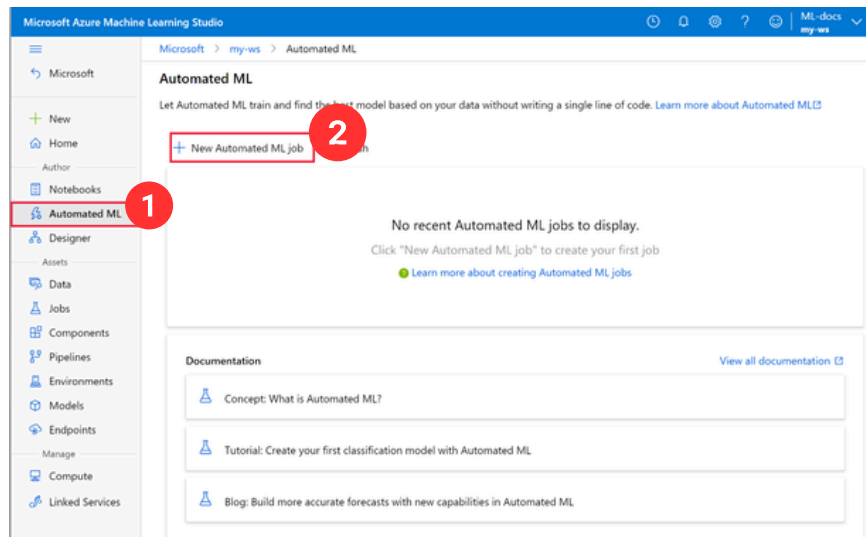


# Azure Machine Learning - AutoML training for Tabular Data [🔗](#)

Automated machine learning (AutoML) is a process that selects the best machine learning algorithm for your specific data, allowing you to generate models quickly. We'll use AutoML to quickly identify the best classification model for our risk dataset.

1. Navigate to the left pane. Select **Automated ML** under the **Authoring** section.

2. Select **+ New automated ML job** and populate the form



3. Specify an **Experiment Name**

4. Select **Classification** as the 'task type'

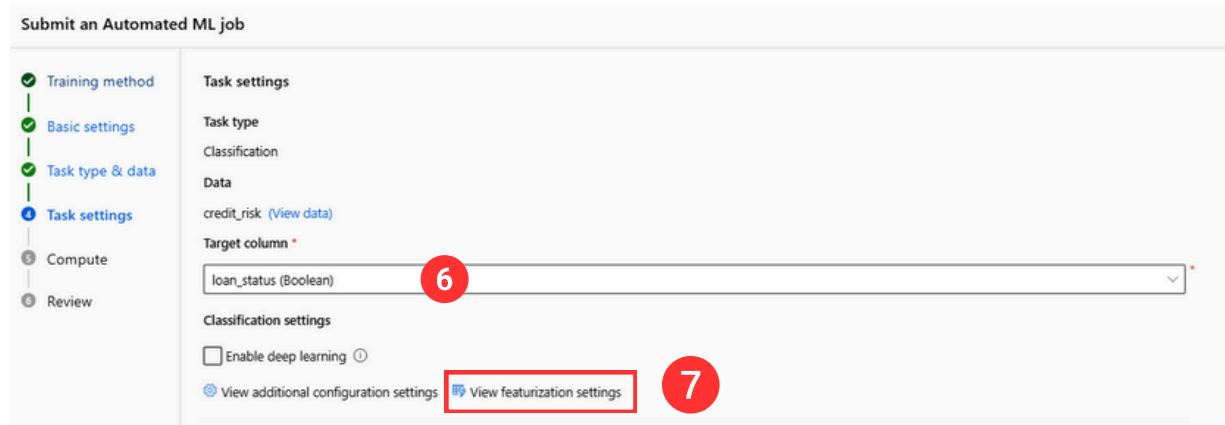
5. Select the existing **data** asset created earlier in the guide.

| Name        | Type  |
|-------------|-------|
| credit_risk | Table |

## Requirements for training data:

- Data must be in tabular form.
- The value you want to predict (target column) must be present in the data.

6. Select the **Target column** (value you want to predict)



Submit an Automated ML job

Training method  
Basic settings  
Task type & data  
Task settings  
Compute  
Review

Task settings

Task type  
Classification

Data  
credit\_risk (View data)

Target column \*

loan\_status (Boolean)

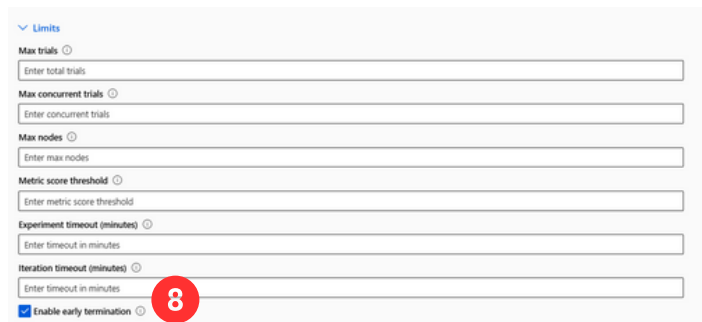
Classification settings

☐ Enable deep learning ⓘ

☒ View additional configuration settings ⓘ [View featurization settings](#) ⓘ

7. (Optional) Enable Automatic featurization in the Additional configuration settings form to apply default techniques. In View featurization settings, you can customize these defaults. Learn how to [customize featurizations](#).

8. Select **Enable-early termination** to end the job if the score is not improving in the short term.



Limits

Max trials ⓘ  
Enter total trials

Max concurrent trials ⓘ  
Enter concurrent trials

Max nodes ⓘ  
Enter max nodes

Metric score threshold ⓘ  
Enter metric score threshold

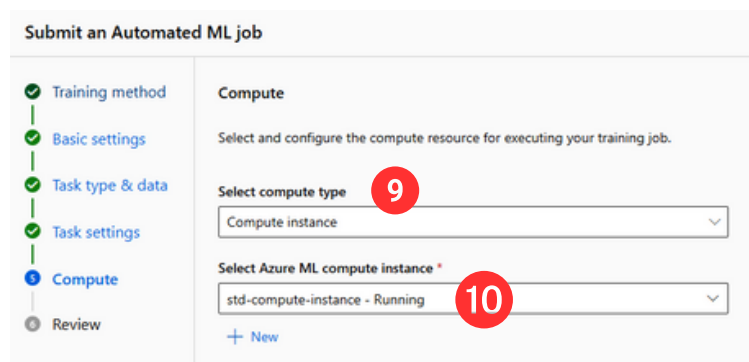
Experiment timeout (minutes) ⓘ  
Enter timeout in minutes

Iteration timeout (minutes) ⓘ  
Enter timeout in minutes

☒ Enable early termination ⓘ

9. Select **Compute instance** for compute type, in order to use the compute created earlier in this guide

10. Select the created Azure ML compute instance.



Submit an Automated ML job

Training method  
Basic settings  
Task type & data  
Task settings  
Compute  
Review

Compute

Select and configure the compute resource for executing your training job.

Select compute type

Compute instance

Select Azure ML compute instance \*

std-compute-instance - Running

[+ New](#)

Submit training job

Submit training job

# Azure Machine Learning - View Job Details

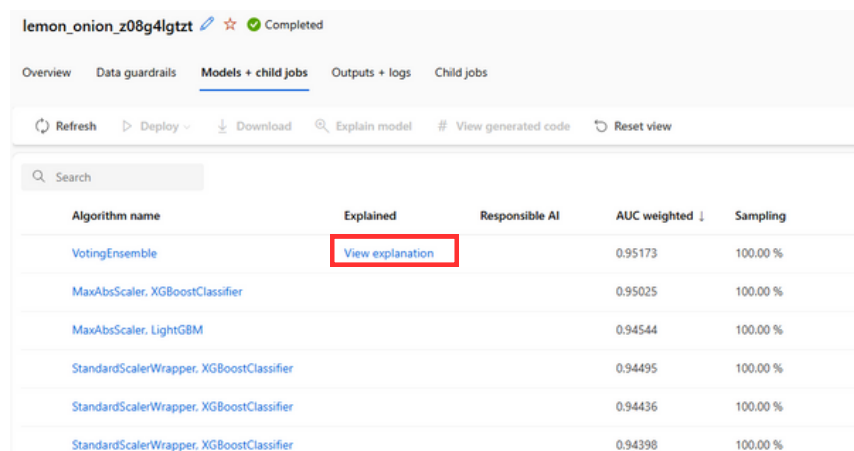


The Job Detail screen opens to the Overview tab, displaying a summary of the experiment job and a summary for the 'Best model' - top-performing model that is determined based on these evaluation metrics.



| Best model summary |                                                |
|--------------------|------------------------------------------------|
| Algorithm name     | VotingEnsemble                                 |
| Ensemble details   | <a href="#">View ensemble details</a>          |
| AUC weighted       | 0.95173 <a href="#">View all other metrics</a> |
| Sampling           | 100.00 %                                       |
| Registered models  | No registration yet                            |
| Deploy status      | No deployment yet                              |

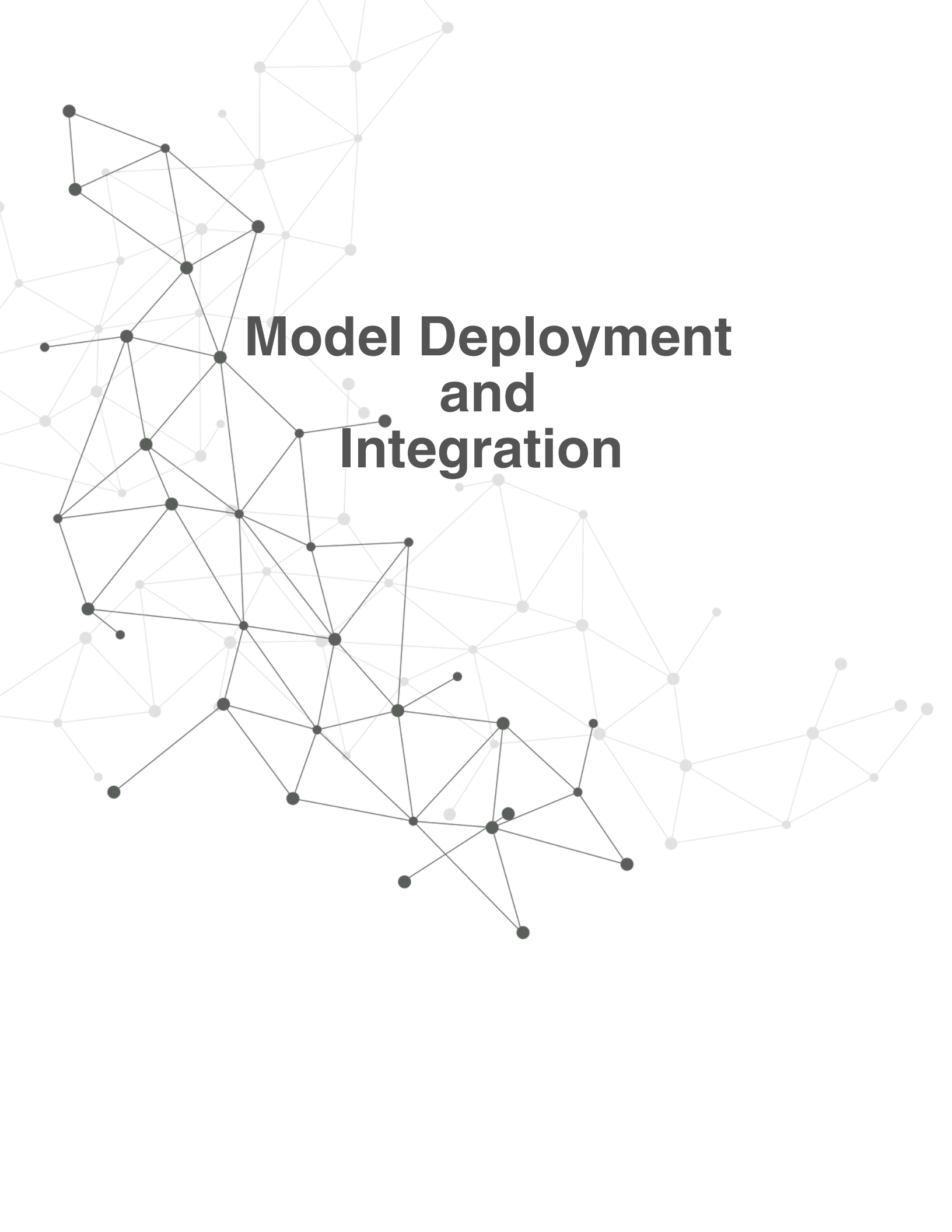
The **Models** tab lists the models created, ordered by their metric scores. By default, the highest-scoring model based on the chosen metric appears at the top. As the training job progresses, newly tested models are added to the list.



| lemon_onion_z08g4lgtzt    Completed                                                                                                                                                                                                                                                                                           |                                  |                     |                |            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|---------------------|----------------|------------|
| Overview                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data guardrails                  | Models + child jobs | Outputs + logs | Child jobs |
|  Refresh  Deploy  Download  Explain model  View generated code  Reset view |                                  |                     |                |            |
| <input type="text" value="Search"/>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                  |                     |                |            |
| Algorithm name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Explained                        | Responsible AI      | AUC weighted ↓ | Sampling   |
| VotingEnsemble                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <a href="#">View explanation</a> |                     | 0.95173        | 100.00 %   |
| MaxAbsScaler, XGBoostClassifier                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                  |                     | 0.95025        | 100.00 %   |
| MaxAbsScaler, LightGBM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                  |                     | 0.94544        | 100.00 %   |
| StandardScalerWrapper, XGBoostClassifier                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                  |                     | 0.94495        | 100.00 %   |
| StandardScalerWrapper, XGBoostClassifier                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                  |                     | 0.94436        | 100.00 %   |
| StandardScalerWrapper, XGBoostClassifier                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                  |                     | 0.94398        | 100.00 %   |

**Explanations** provides insights into how models make predictions, highlighting feature importance and model behavior through global and local explanations. It supports interpretability techniques like SHAP and LIME, integrated with Azure ML to enhance transparency, compliance, and trust in AI systems.



An abstract network diagram featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, with some in black and others in grey. The lines are thin and grey, creating a dense, interconnected pattern that fills the background. The overall structure is non-linear and organic, suggesting a complex system or network.

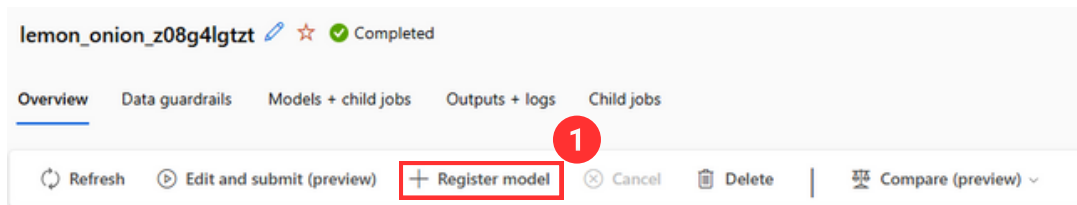
# **Model Deployment and Integration**



# Azure Machine Learning - Register Model from Job [🔗](#)

It is recommended to register your model and environment before deployment to Azure. This allows you to specify their registered names and versions during deployment, enabling you to reuse these assets without re-uploading them each time, which enhances reproducibility and traceability.

1. From your recent model run, click **+ Register model**.



2. Select MLflow as the **Model type**.

3. Use the best\_model **Job output**.

## Register model from a job output

4 Review

3 Model settings

2 Select output

1 Select job

Select output

Specify the corresponding output to register the model.

Model type \* ① 2

MLflow

Job output \* ③ 3

best\_model (azureml\_lemon\_onion\_z08g4lgtzt\_40\_output\_mlflow\_log\_model...

## Model registries with MLflow [🔗](#)

Azure Machine Learning integrates with MLflow for model management, offering a streamlined approach to support the entire model lifecycle for users accustomed to the MLflow client.

4. Select **Next**.

5. Enter a friendly **Name** for the model; for example, risk-model-1.

6. Leaving **Version** 'blank' will autoincrement the registered model version number.

7. Select **Next**, and then **Register** to complete registration.

| Name         | ☆ | Version | Type   | Source         | Experiment  |
|--------------|---|---------|--------|----------------|-------------|
| risk-model-1 |   | 1       | MLFLOW | This workspace | credit_risk |

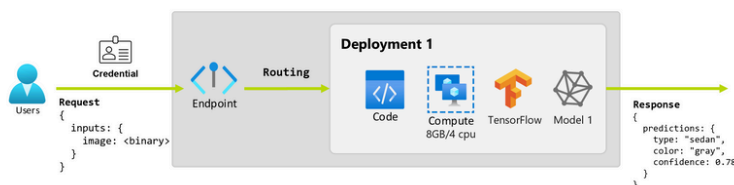
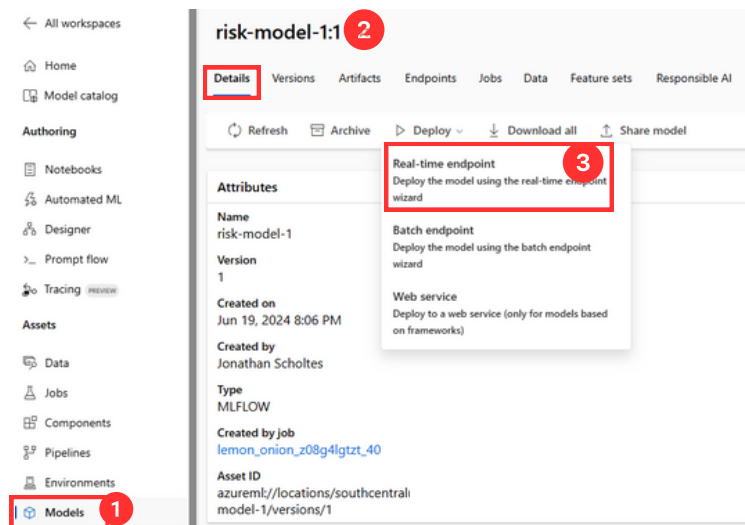
# Azure Machine Learning - Model Deployment



Azure Machine Learning enables real-time inferencing by deploying models to online endpoints, allowing new input data to generate outputs. These outputs, often called "predictions," can also include results for tasks like classification and clustering.

## Deploy Real-time Endpoint from Registered Model

1. Select **Models** from the Navigation
2. Select the **Register** model to deploy (risk-model-1(version 1))
3. Select **Real-time endpoint** from **Deploy**.
4. Select the compute for the deployment by specifying **Virtual machine** type and **Instance count**.
5. Specify **Endpoint Name** and **Deployment Name**



A deployment in Azure Machine Learning hosts the resources for inferencing models or components within an endpoint, which can contain multiple deployments with independent assets and tailored resource consumption, facilitated by a routing mechanism for directing requests

## When to Use Serverless APIs, Online Endpoints, and Batch Endpoints

### Serverless API Endpoints:

Use serverless API endpoints for real-time inferencing with large foundational models, either off-the-shelf or fine-tuned. Ideal for models available for serverless deployment, offering quota-less usage without the need to customize the inferencing stack.

### Online Endpoints:

Operationalize models for real-time inference with low-latency, synchronous requests. Best for models not supported in serverless API endpoints, with low-latency requirements, quick response times, and inputs fitting in HTTP payloads, allowing for scalable request handling.

### Batch Endpoints:

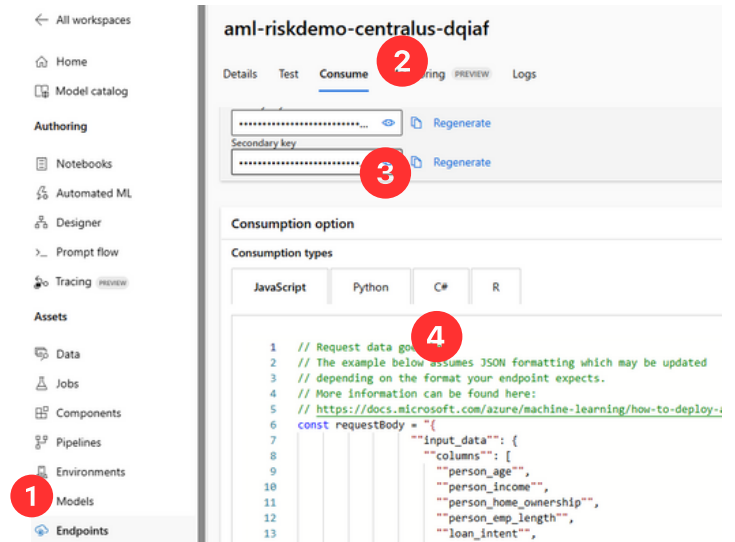
Deploy models or pipelines for long-running, asynchronous inference. Suitable for expensive, time-consuming models, operationalizing ML pipelines, handling large datasets in storage, and leveraging parallelization without low-latency needs.



# Application Azure Machine Learning Consumption

Consuming an Azure Machine Learning online endpoint via a web application enables real-time predictions, enhancing responsiveness and user experience. It offers scalability, ease of integration, robust security, and centralized model management, ensuring efficient and secure operation

1. Navigate to the **Endpoints** section and find your deployed online endpoint.
2. Select **Consume**.
3. Note down the **endpoint URL** and any necessary **authentication keys** or tokens.
4. Copy the code from the **Consumption options**.



Integrate real-time predictions into your application by using the endpoint URL, authentication keys, and the provided example code

## Example

