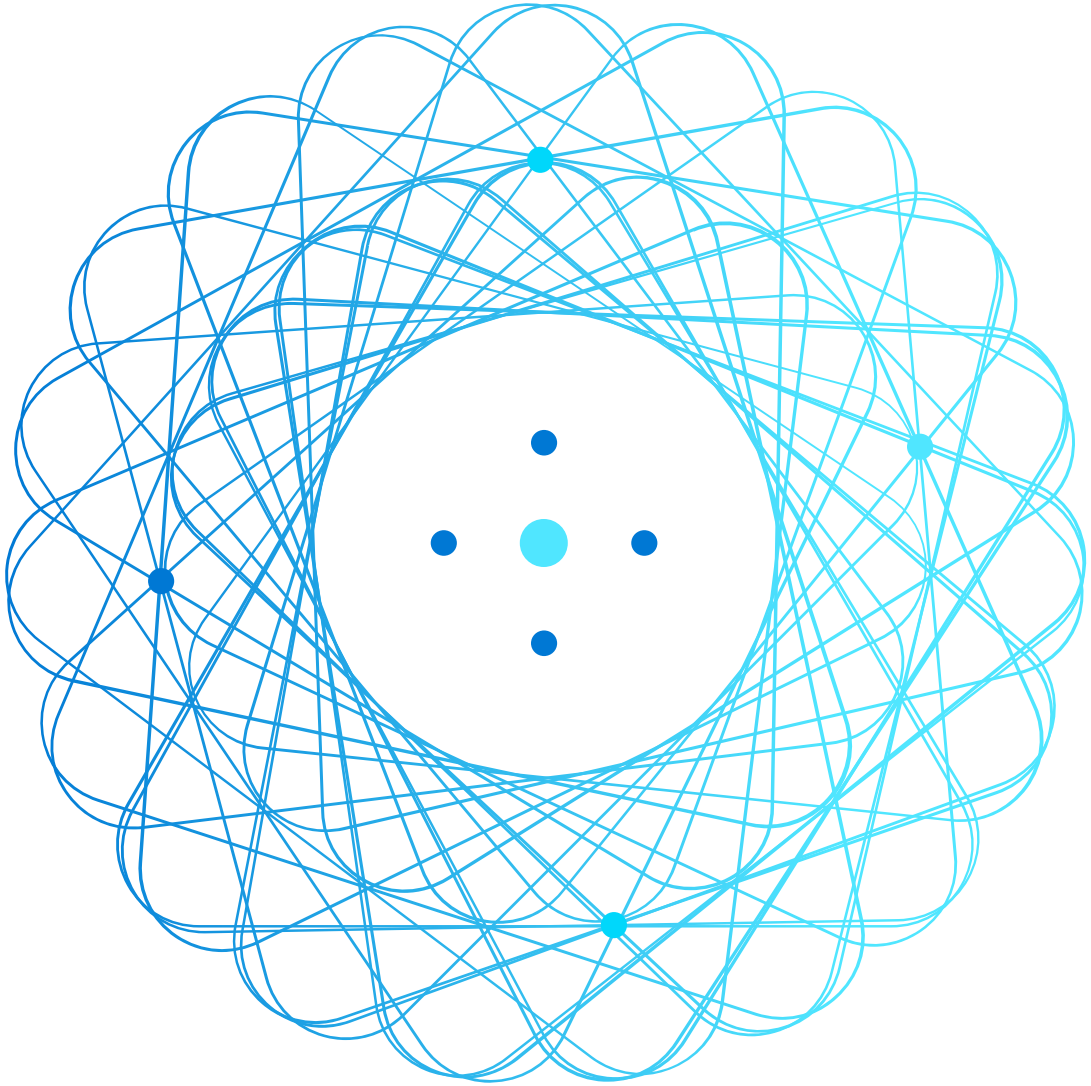
Microsoft Azure

Deploy and consume models with Azure Machine Learning

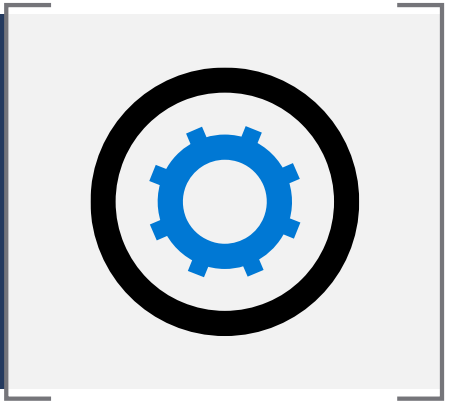


Module Agenda

Deploy a model to a managed online endpoint

Deploy a model to a batch endpoint

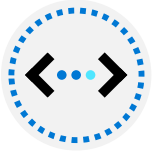
Deploy a model to a managed online endpoint



Explore managed online endpoints

Real-time predictions: To get real-time predictions, you can deploy a model to an endpoint

Managed online endpoint: Within Azure Machine Learning, there are two types of online endpoints: Managed online endpoints and Kubernetes online endpoints



Deploy your model: After you create an endpoint in the Azure Machine Learning workspace, you can deploy a model to that endpoint

Blue/green deployment: One endpoint can have multiple deployments. One approach is the blue/green deployment.

Create an endpoint: To create an online endpoint, you'll use the `ManagedOnlineEndpoint` class

Deploy your MLflow model to a managed online endpoint

Deploy an MLflow model to an endpoint

- When you deploy an MLflow model to a managed online endpoint, you don´t need to have the scoring script and environment.
- To deploy an MLflow model, you must have model files stored on a local path or with a registered model.
- Next to the model, you also need to specify the compute configuration for the deployment:
  - **instance_type:** Virtual machine (VM) size to use.
  - **instance_count:** Number of instances to use.

Deploy a model to a managed online endpoint

Deploy a model to an endpoint

- **Create the scoring script**

  The scoring script needs to include two functions:

  - `init()`: Called when the service is initialized
  - `run()`: Called when new data is submitted to the service

- **Create an environment**

  Your deployment requires an execution environment in which to run the scoring script.

- **Create the deployment**

  When you have your model files, scoring script, and environment, you can create the deployment.

To deploy a model, you must have:

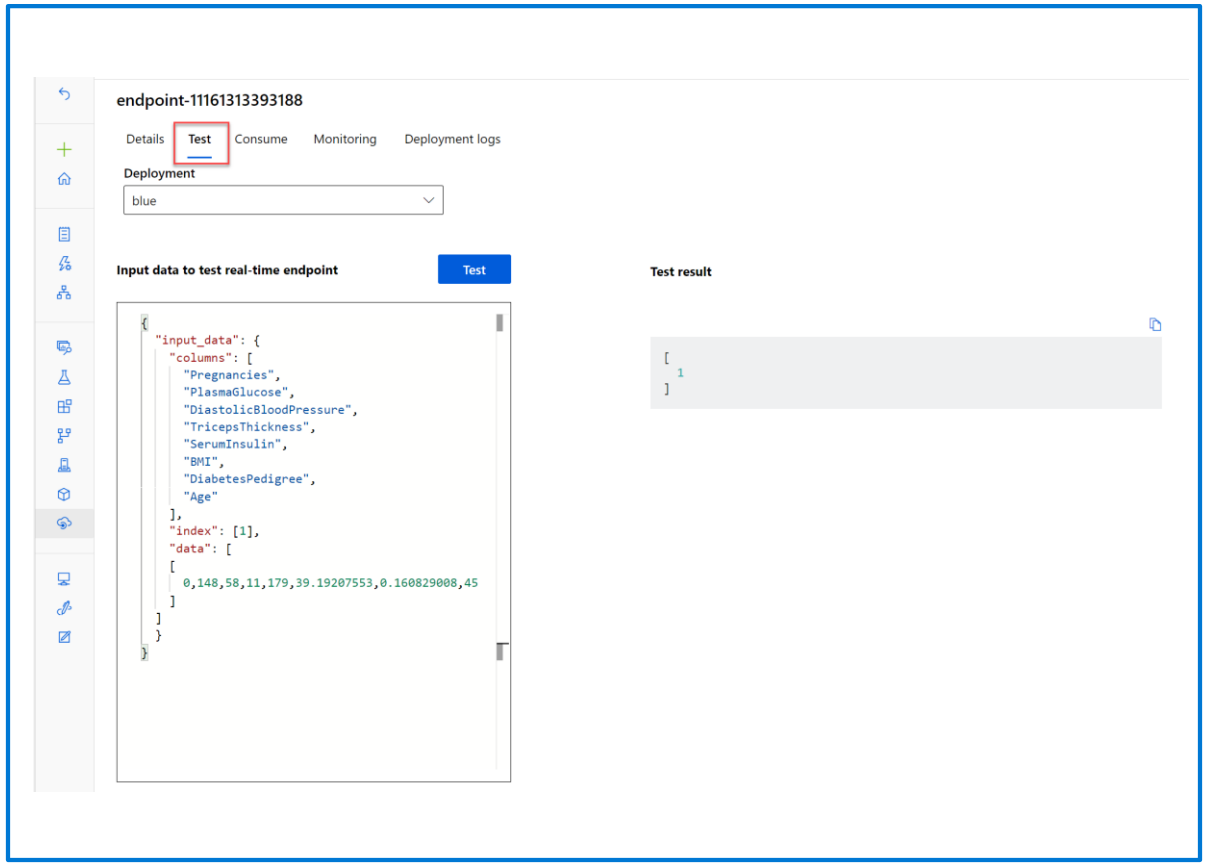
- Model files stored on local path or registered model
- A scoring script
- An execution environment

The model files can be logged and stored when you train a model.

Test managed online endpoints

Use the Azure Machine Learning studio

- You can list all endpoints in the Azure Machine Learning studio, by navigating to the **Endpoints** page. In the **Real-time endpoints** tab, all endpoints are shown.
- You can select an endpoint to review its details and deployment logs.
- Additionally, you can use the studio to test the endpoint.



Use the Azure Machine Learning Python SDK

For testing, you can also use the Azure Machine Learning Python SDK to invoke an endpoint.

Send data to the deployed model in JSON format with the following structure:

JSON

```
{
  "data":[
     [0.1,2.3,4.1,2.0], // 1st case
     [0.2,1.8,3.9,2.1],  // 2nd case,
     ...
  ]
}
```

The response from the deployed model is a JSON collection with a prediction for each case that was submitted in the data. The following code sample invokes an endpoint and displays the response:

Python

```
# test the blue deployment with some sample data
response = ml_client.online_endpoints.invoke(
    endpoint_name=online_endpoint_name,
    deployment_name="blue",
    request_file="sample-data.json",
)

if response[1]=='1’:
    print("Yes")
else:
    print ("No")
```

Exercise - Deploy an MLflow model to an online endpoint

Task 1: Create a managed online endpoint

Task 2: Deploy an MLflow model

Task 3: Test the endpoint

Instructions

Follow these instructions to complete the exercise:

- View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.
- Complete the **Deploy a model to a managed online endpoint** exercise.

Knowledge check

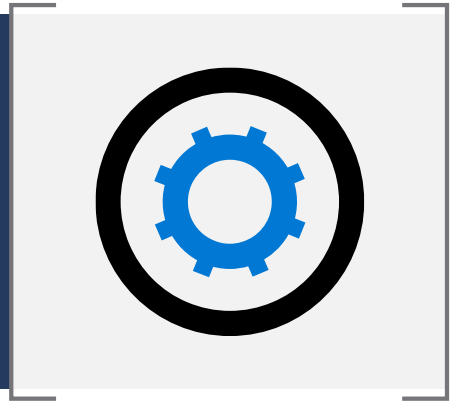
You've trained a model using the Python SDK for Azure Machine Learning. You want to deploy the model to get real-time predictions. You want to manage the underlying infrastructure used by the endpoint. What kind of endpoint should you create?

- ❑ A managed online endpoint.
- ❑ A batch endpoint.
- ☑ A Kubernetes online endpoint.

You're deploying a model as a real-time inferencing service. What functions must the scoring script for the deployment include?

- ❑ `main()` and `score()`
- ❑ `base()` and `train()`
- ☑ `init()` and `run()`

Deploy a model to a batch endpoint



Understand and create batch endpoints (1/4)

Batch predictions

To get batch predictions, you can deploy a model to an endpoint

An **endpoint** is an HTTPS endpoint that you can call to trigger a batch scoring job

The advantage of such an endpoint is that you can trigger the batch scoring job from another service, such as Azure Synapse Analytics or Azure Databricks

Whenever the endpoint is invoked, a batch scoring job is submitted to the Azure Machine Learning workspace

Understand and create batch endpoints (2/4)

Create a batch endpoint

- To deploy a model to a batch endpoint, you'll first have to create the batch endpoint.
- To create a batch endpoint, you'll use the `BatchEndpoint` class. Batch endpoint names need to be unique within an Azure region.

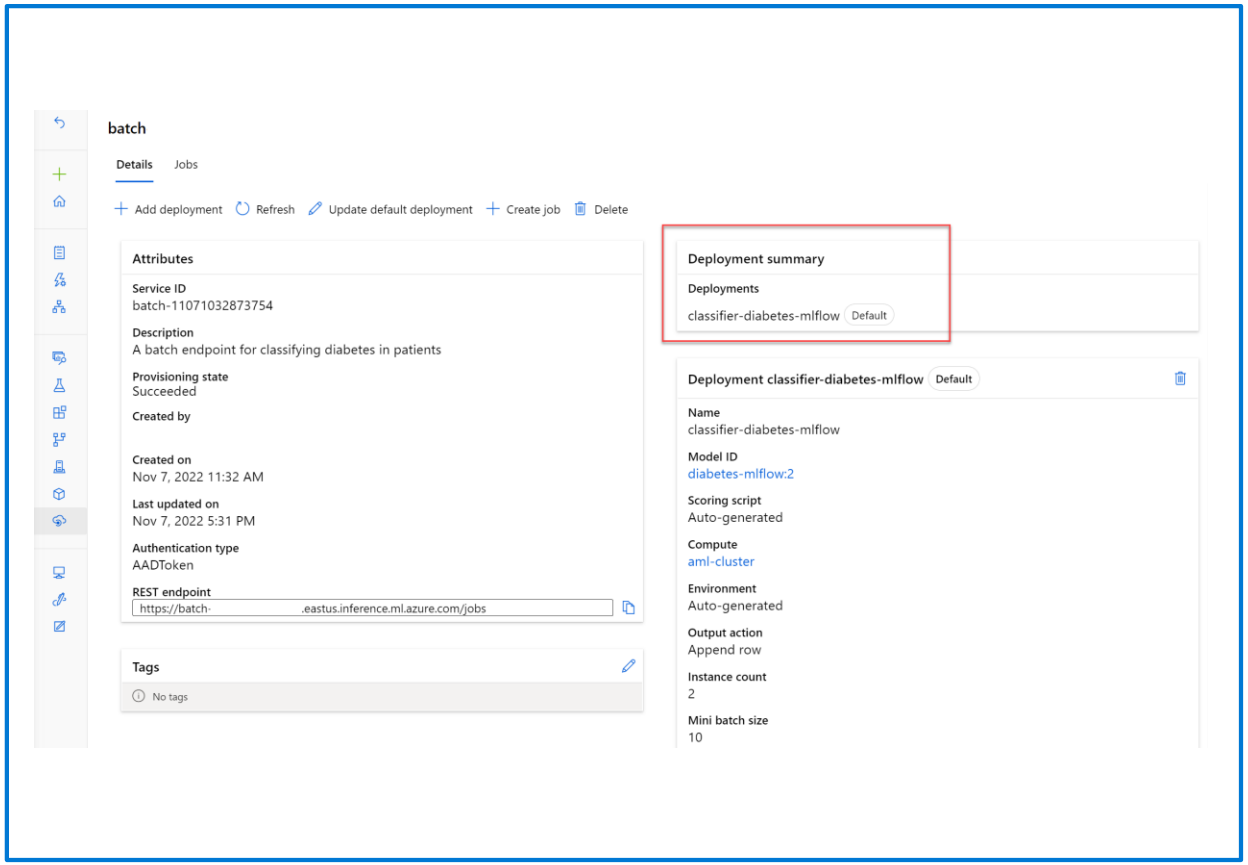
Python

```
# create a batch endpoint
endpoint = BatchEndpoint(
    name="endpoint-example",
    description="A batch endpoint",
)


ml_client.batch_endpoints.begin_create_or_update(endpoint)
```

Understand and create batch endpoints (3/4)

Deploy a model to a batch endpoint

You can deploy multiple models to a batch endpoint. Whenever you call the batch endpoint, which triggers a batch scoring job, the **default deployment** will be used unless specified otherwise.



Understand and create batch endpoints (4/4)

Use compute clusters for batch deployments

- The ideal compute to use for batch deployments is the Azure Machine Learning compute cluster.
- If you want the batch scoring job to process the new data in parallel batches, you need to provision a compute cluster with more than one maximum instances.
- To create a compute cluster, you can use the `AMLCompute` class.

Python

```
from azure.ai.ml.entities import AmlCompute

cpu_cluster = AmlCompute(
    name="aml-cluster",
    type="amlcompute",
    size="STANDARD_DS11_V2",
    min_instances=0,
    max_instances=4,
    idle_time_before_scale_down=120,
    tier="Dedicated",
)

cpu_cluster =
ml_client.compute.begin_create_or_update(cpu_cluster)
```

Deploy your MLflow model to a batch endpoint (1/2)

Register an MLflow model

- For no-code deployment, an MLflow model needs to be registered in the Azure Machine Learning workspace before you can deploy it to a batch endpoint.
- To register an MLflow model, you'll use the `Model` class, while specifying the model type to be `MLFLOW_MODEL`.
- To register the model with the Python SDK, you can use the following code:

Python

```
from azure.ai.ml.entities import Model
from azure.ai.ml.constants import AssetTypes

model_name = 'mlflow-model'
model = ml_client.models.create_or_update(
    Model(name=model_name, path='./model',
    type=AssetTypes.MLFLOW_MODEL)
)
```

Deploy your MLflow model to a batch endpoint (2/2)

Deploy an MLflow model to an endpoint

- To deploy an MLflow model to a batch endpoint, you'll use the `BatchDeployment` class.
- When you deploy a model, you'll need to specify how you want the batch scoring job to behave.
- When you configure the model deployment, you can specify:
  - `instance_count`
  - `max_concurrency_per_instance`
  - `mini_batch_size`
  - `output_action`
  - `output_file_name`

Deploy a custom model to a batch endpoint

If you want to deploy a model to a batch endpoint without using the MLflow model format, you need to create the scoring script and environment.

Create the scoring script: The scoring script is a file that reads the new data, loads the model, and performs the scoring.

Create an environment: Your deployment requires an execution environment in which to run the scoring script. Any dependency your code requires should be included in the environment.

Configure and create the deployment: Finally, you can configure and create the deployment with the `BatchDeployment` class.

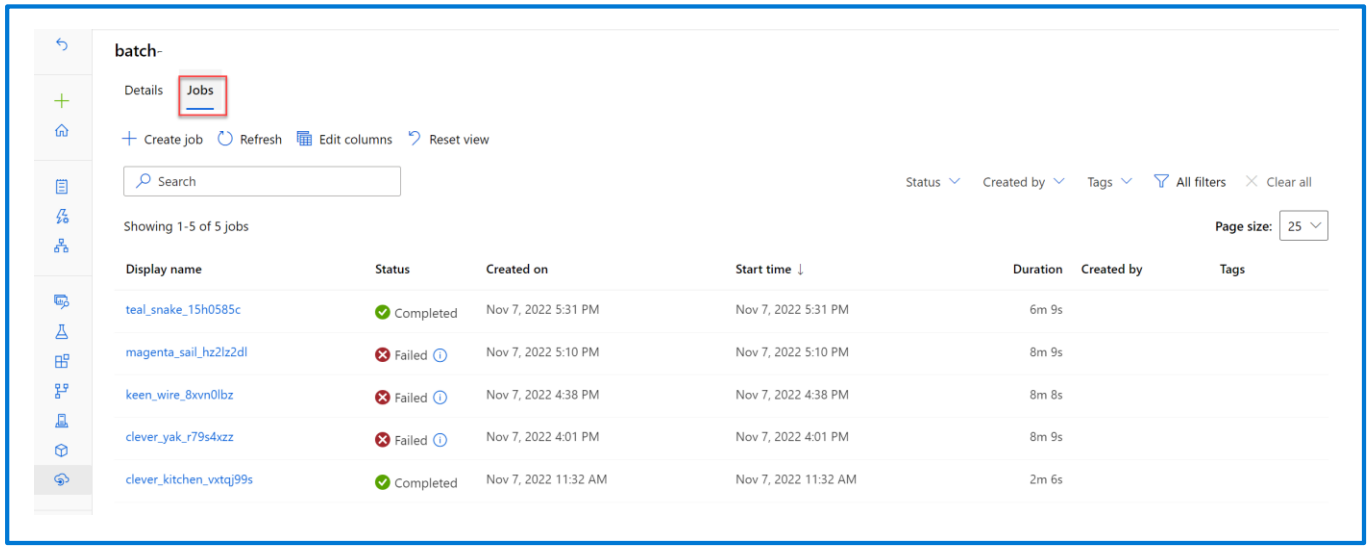
Invoke and troubleshoot batch endpoints (1/2)

Trigger the batch scoring job

- To prepare data for batch predictions, you can register a folder as a data asset in the Azure Machine Learning workspace.
- You can then use the registered data asset as input when invoking the batch endpoint with the Python SDK:

Python

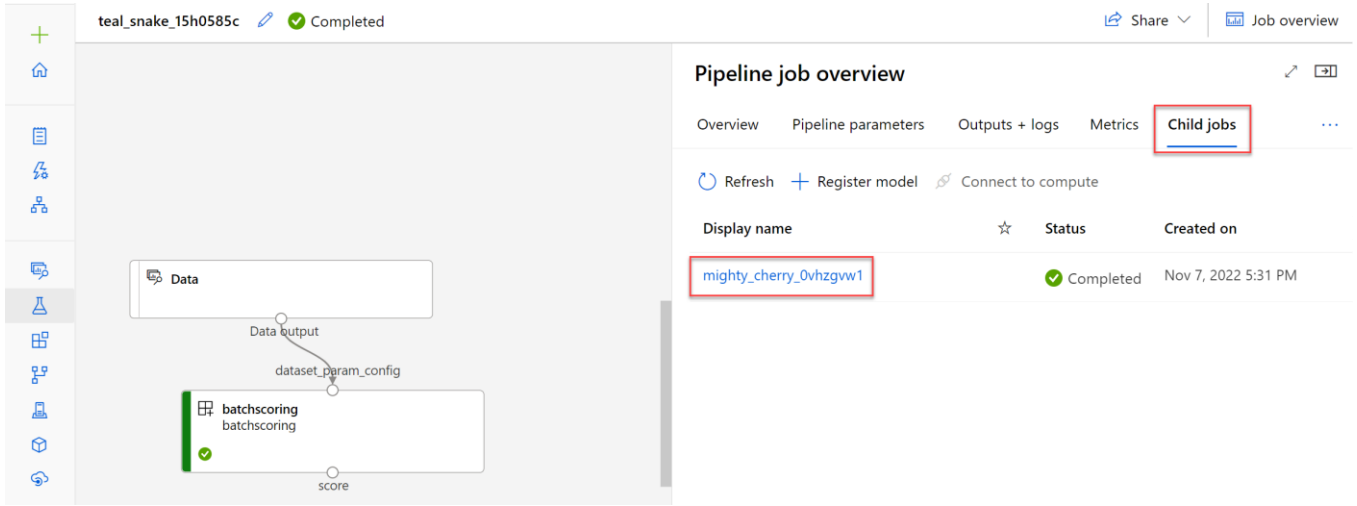
```
from azure.ai.ml import Input
from azure.ai.ml.constants import AssetTypes

input = Input(type=AssetTypes.URI_FOLDER, path="azureml:new-data:1")

job = ml_client.batch_endpoints.invoke(
    endpoint_name=endpoint.name,
    input=input)
```



Invoke and troubleshoot batch endpoints (2/2)

Troubleshoot a batch scoring job

The batch scoring job runs as a *pipeline job*. If you want to troubleshoot the pipeline job, you can review its details and the outputs and logs of the pipeline job itself.



Exercise - Deploy an MLflow model to a batch endpoint

Task 1: Create a batch endpoint

Task 2: Deploy an MLflow model to the endpoint

Task 3: Invoke the endpoint

Instructions

Follow these instructions to complete the exercise:

- View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.
- Complete the **Deploy a model to a batch endpoint** exercise.

Knowledge check

You are creating a batch endpoint that you want to use to predict new values for a large volume of data files. You want the pipeline to run the scoring script on multiple nodes and collate the results. What output action should you choose for the deployment?

- ❑ summary_only
- ☑ append_row
- ❑ concurrency

You have multiple models deployed to a batch endpoint. You invoke the endpoint without indicating which model you want to use. What deployed model will do the actual batch scoring?

- ❑ The latest version of the deployed model.
- ❑ The latest deployed model.
- ☑ The default deployed model.

Microsoft Azure