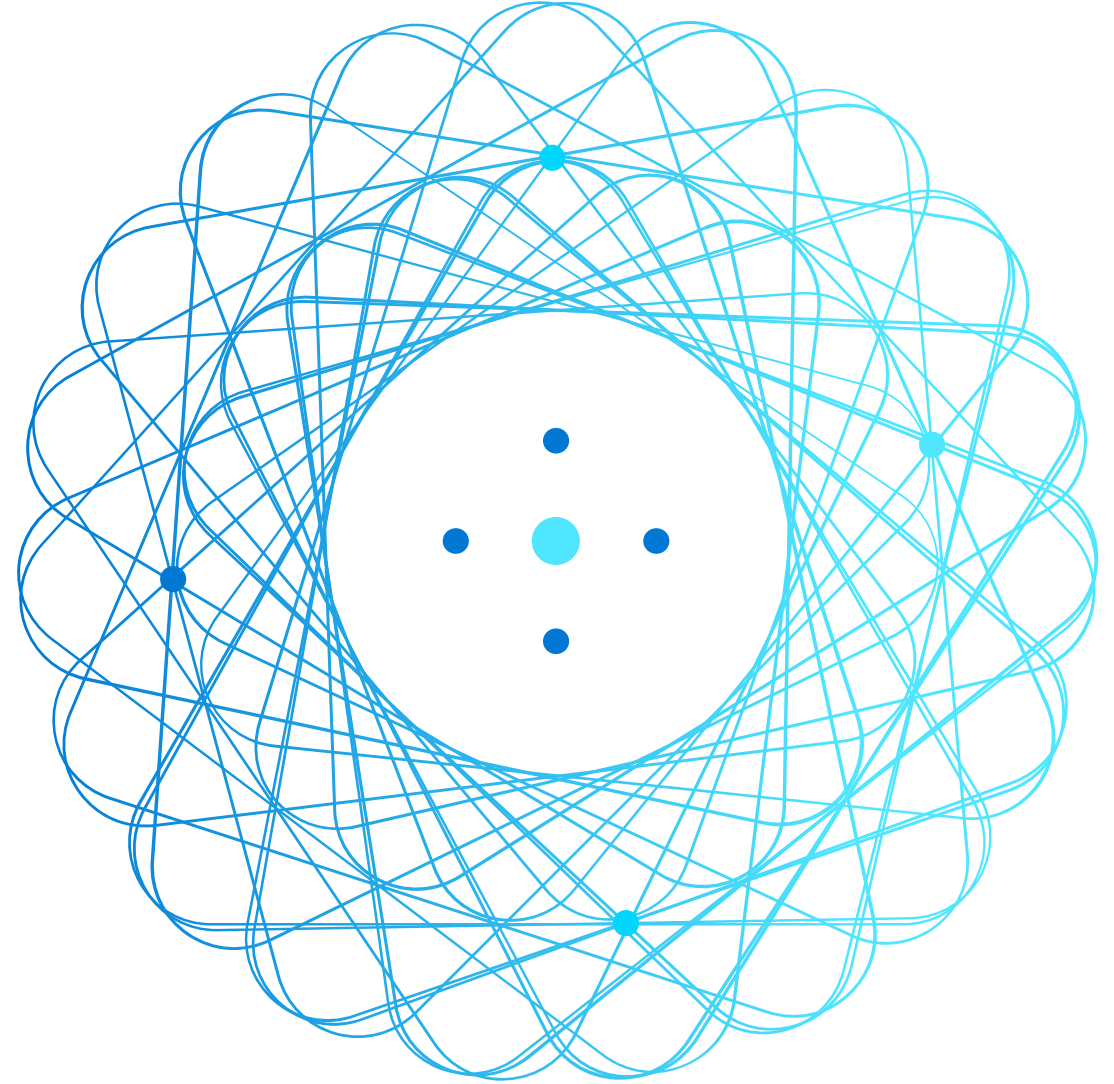


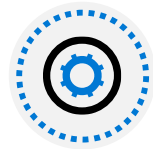
# Deploy and consume models with Azure Machine Learning



# Module Agenda

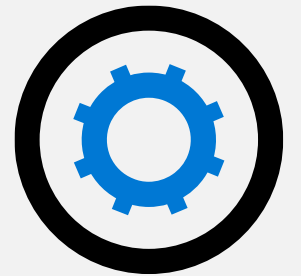


Deploy a model to a managed online endpoint



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:

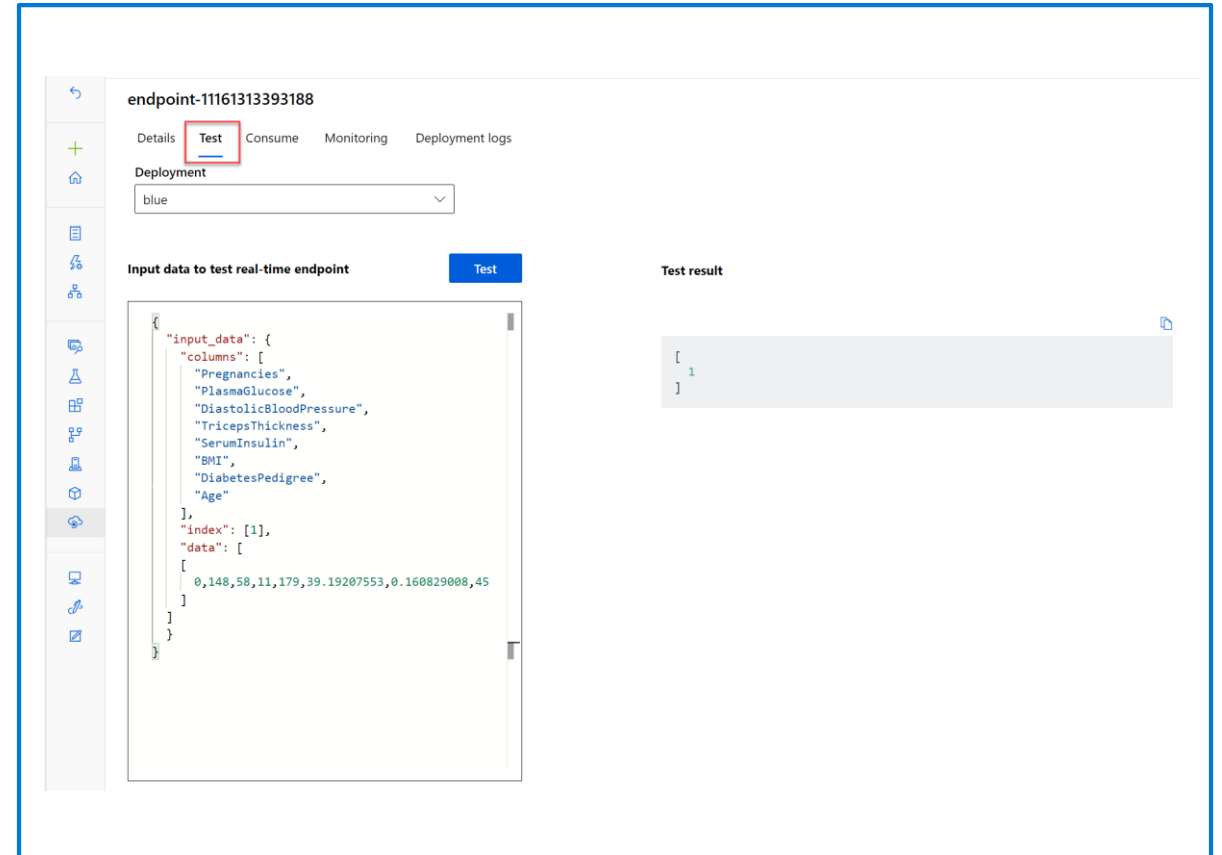
- Model files stored on local path or registered model
- A scoring script
- An execution environment

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.



The screenshot displays the Azure Machine Learning studio interface for testing a managed online endpoint. The endpoint name is 'endpoint-11161313393188'. The 'Test' tab is selected and highlighted with a red box. The 'Deployment' dropdown menu is set to 'blue'. The 'Input data to test real-time endpoint' section contains a JSON object with the following structure:

```
{
  "input_data": {
    "columns": [
      "Pregnancies",
      "PlasmaGlucose",
      "DiastolicBloodPressure",
      "TricepsThickness",
      "SerumInsulin",
      "BMI",
      "DiabetesPedigree",
      "Age"
    ],
    "index": [1],
    "data": [
      [
        0, 148, 58, 11, 179, 39.19207553, 0.160829008, 45
      ]
    ]
  }
}
```

The 'Test result' section shows a single value '1'.

# 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")
```

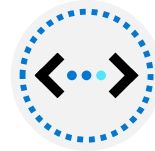


# Exercise - Deploy an MLflow model to an online endpoint

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?

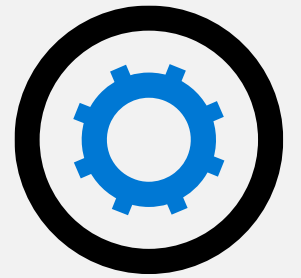
- A managed online endpoint.
- A batch endpoint.
- A Kubernetes online endpoint.



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

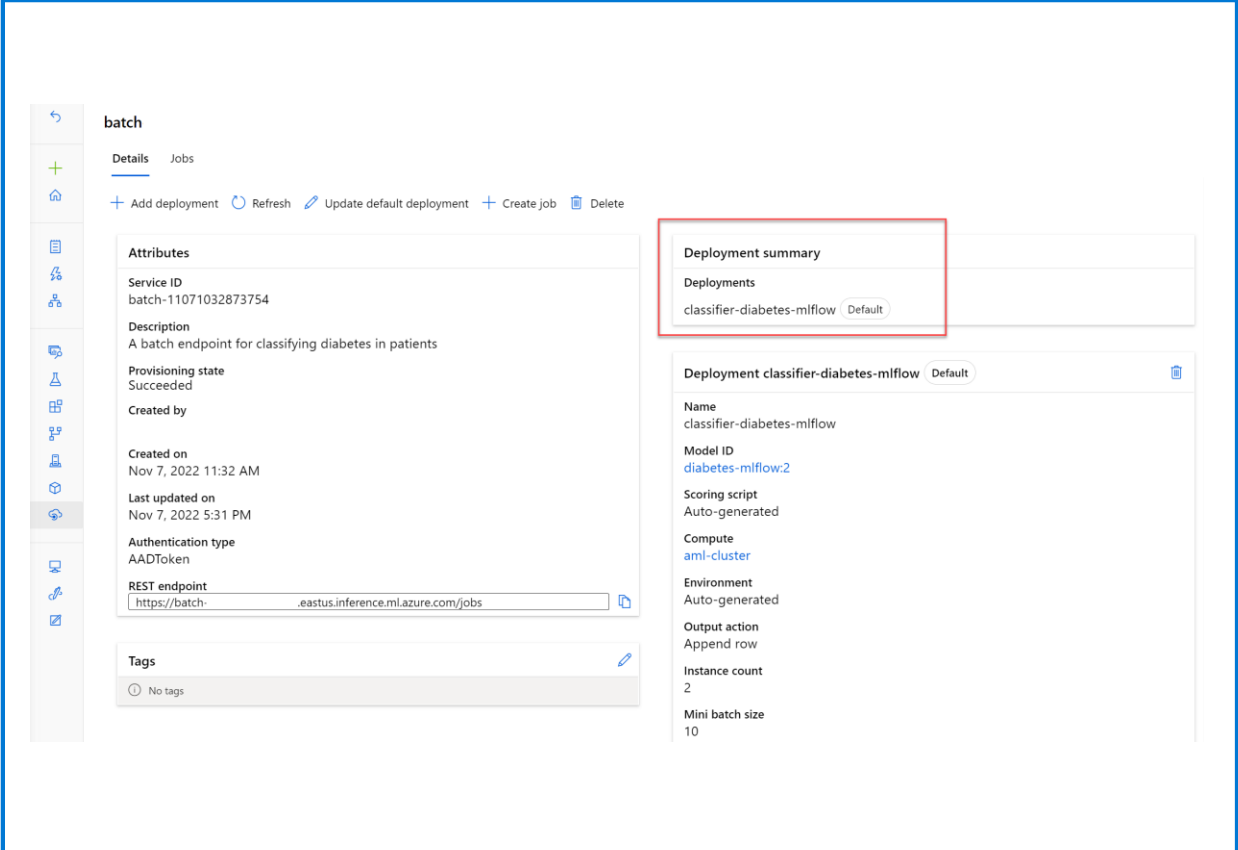
```
# create a batch endpoint
endpoint = BatchEndpoint(
    name="endpoint-example",
    description="A batch endpoint",
)

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.



The screenshot displays the 'batch' endpoint details in the Azure Machine Learning interface. The page is titled 'batch' and includes a navigation bar with 'Details' and 'Jobs' tabs. Below the navigation bar are action buttons: '+ Add deployment', 'Refresh', 'Update default deployment', '+ Create job', and 'Delete'. The main content area is divided into several sections:

- Attributes:** Service ID (batch-11071032873754), Description (A batch endpoint for classifying diabetes in patients), Provisioning state (Succeeded), Created by, Created on (Nov 7, 2022 11:32 AM), Last updated on (Nov 7, 2022 5:31 PM), Authentication type (AADToken), and REST endpoint (https://batch-...eastus.inference.ml.azure.com/jobs).
- Tags:** No tags.
- Deployment summary:** A red box highlights this section, which shows 'Deployments' and a 'classifier-diabetes-mlflow' deployment with a 'Default' tag.
- Deployment classifier-diabetes-mlflow:** A detailed view of the selected deployment, showing Name (classifier-diabetes-mlflow), Model ID (diabetes-mlflow:2), Scoring script (Auto-generated), Compute (aml-cluster), Environment (Auto-generated), Output action (Append row), Instance count (2), and Mini batch size (10).

# 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")
```

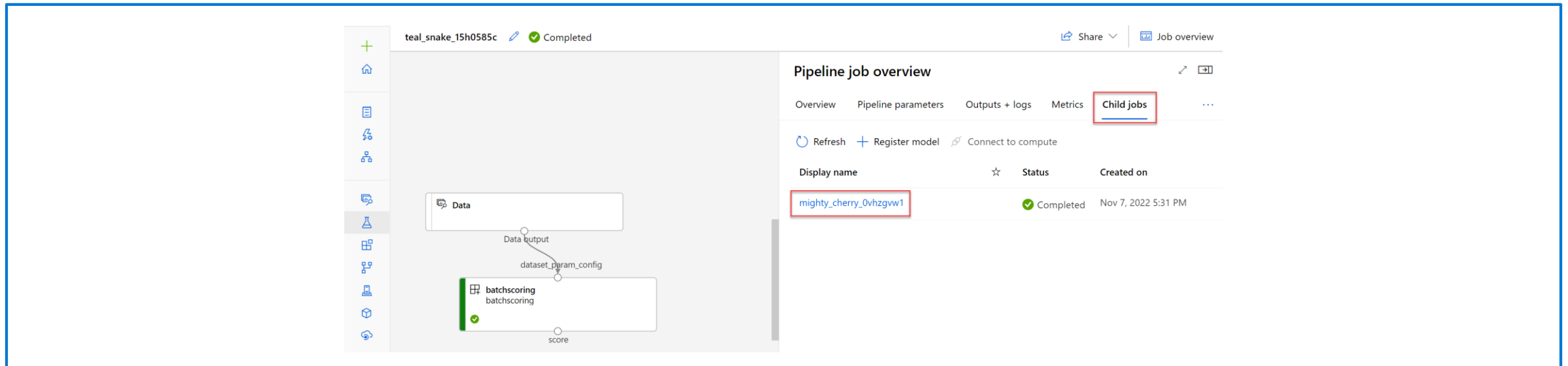
```
job = ml_client.batch_endpoints.invoke(
    endpoint_name=endpoint.name,
    input=input)
```

Display name	Status	Created on	Start time ↓	Duration	Created by	Tags
teal_snake_15h0585c	Completed	Nov 7, 2022 5:31 PM	Nov 7, 2022 5:31 PM	6m 9s		
magenta_sail_hz2lz2dl	Failed	Nov 7, 2022 5:10 PM	Nov 7, 2022 5:10 PM	8m 9s		
keen_wire_8xvn0lbz	Failed	Nov 7, 2022 4:38 PM	Nov 7, 2022 4:38 PM	8m 8s		
clever_yak_r79s4xzz	Failed	Nov 7, 2022 4:01 PM	Nov 7, 2022 4:01 PM	8m 9s		
clever_kitchen_vxtqj99s	Completed	Nov 7, 2022 11:32 AM	Nov 7, 2022 11:32 AM	2m 6s		

# 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.



The screenshot displays the Azure ML interface for a pipeline job. The main area shows a pipeline diagram with a 'Data' node, a 'Data output' node, a 'dataset\_param\_config' node, and a 'batchscoring' node. The 'batchscoring' node is highlighted with a green checkmark, indicating it is completed. The right-hand side shows the 'Pipeline job overview' panel, which includes tabs for 'Overview', 'Pipeline parameters', 'Outputs + logs', 'Metrics', and 'Child jobs'. The 'Child jobs' tab is selected and highlighted with a red box. Below the tabs, there are buttons for 'Refresh', '+ Register model', and 'Connect to compute'. A table lists the child jobs, with the first job, 'mighty\_cherry\_0vhzgw1', highlighted with a red box. The table has columns for 'Display name', 'Status', and 'Created on'.

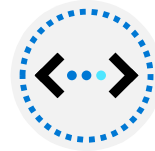
Display name	Status	Created on
mighty_cherry_0vhzgw1	Completed	Nov 7, 2022 5:31 PM

# 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.

