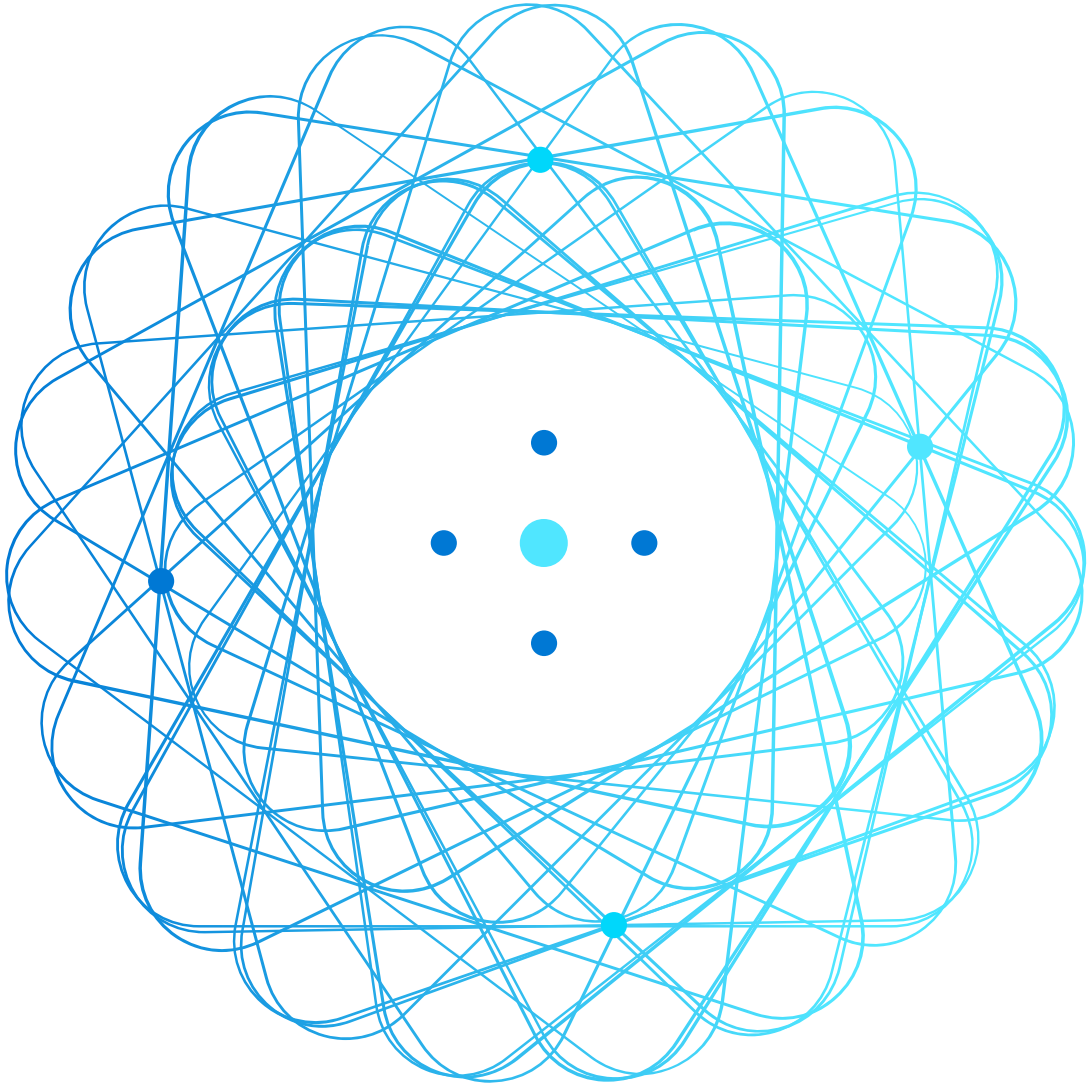
Microsoft Azure

Module 7: Deploying and Consuming Models

Mohammed Arif



Agenda

Real-time Inferencing

Batch Inferencing

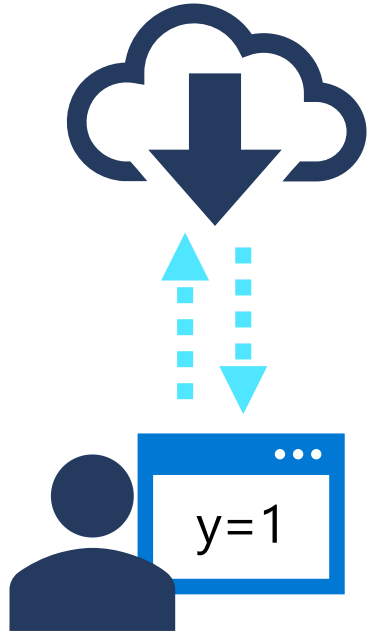
Continuous Integration and Delivery

Real-time Inferencing

What is Real-Time Inferencing?

Immediate prediction from new data

Usually deployed as a web service endpoint



Deploying a Real-Time Inferencing Service

1. **Register a trained model**
2. **Define an Inference Configuration**
   - Create a scoring script (implement **init()** and **run()** functions to load the model and return predictions)
   - Create an environment (use a Conda configuration file)
3. **Define a Deployment Configuration**
   - Create a Compute Target (for example: local, Azure Container Instance, AKS cluster)
4. **Deploy the model as a service**

```
service = Model.deploy(ws, 'my_service', [model], inference_config, deploy_config)
```

Consuming a Real-time Inferencing Service

Use the SDK

```
import json

x_new = [[0.1,2.3,4.1,2.0],[0.2,1.8,3.9,2.1]] # Array of feature vectors
json_data = json.dumps({"data": x_new})
response = service.run(input_data = json_data)
predictions = json.loads(response)
```

Use the REST Endpoint

```
import json
import requests

x_new = [[0.1,2.3,4.1,2.0],[0.2,1.8,3.9,2.1]] # Array of feature vectors
json_data = json.dumps({"data": x_new})
request_headers = { 'Content-Type':'application/json' }
response = requests.post(url=endpoint, data=json_data, headers=request_headers)
predictions = json.loads(response.json())
```

Troubleshooting a Real-Time Inferencing Service

Check the service state

```
print(service.state)
```

Review service logs

```
print(service.get_logs())
```

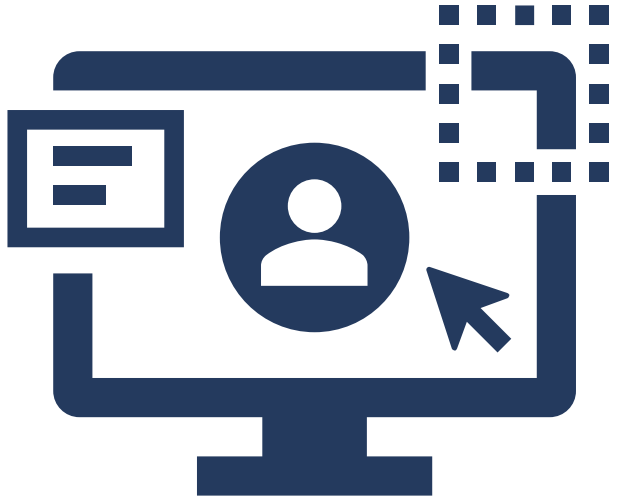
Deploy to a local container

```
deployment_config = LocalWebservice.deploy_configuration(port=8890)
service = Model.deploy(ws, 'test-svc', [model], inference_config, deployment_config)
```

Modify entry script to debug, and then reload to test

```
service.reload()
service.run(input_data=test_sample)
```

Lab: Create a Real-time Inference Service



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Create a real-time inference service** exercise

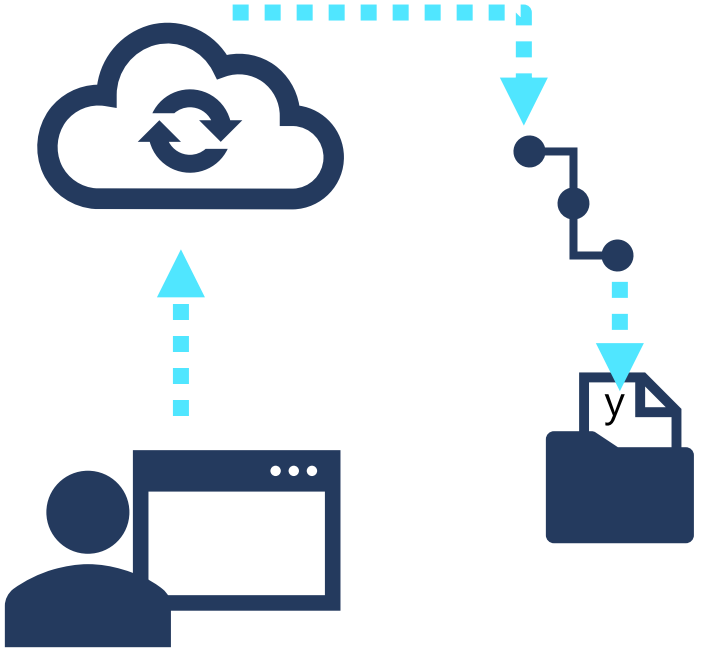
Batch Inferencing

What is Batch Inferencing?

Asynchronous prediction from batched data

Implemented as a pipeline

- Typically using a ParallelRunStep for scalability



Creating a Batch Inferencing Pipeline

1. **Register the model**
2. **Create a scoring script**
   - Implement **init()** and **run(mini_batch)** functions to load the model and return predictions for each mini-batch
3. **Create a pipeline with a ParallelRunStep to run the script**
   - Define a **File** dataset input for the batch data
   - Define a **OutputFileDatasetConfig** reference for the output folder
   - Configure with an **output_action** of "append_row" so all results are collated in *parallel_run_step.txt*.
4. **Retrieve batch predictions from the output**

Publishing a Batch Inferencing Service

Publish the batch pipeline as a REST service

Use the pipeline endpoint to initiate batch inferencing

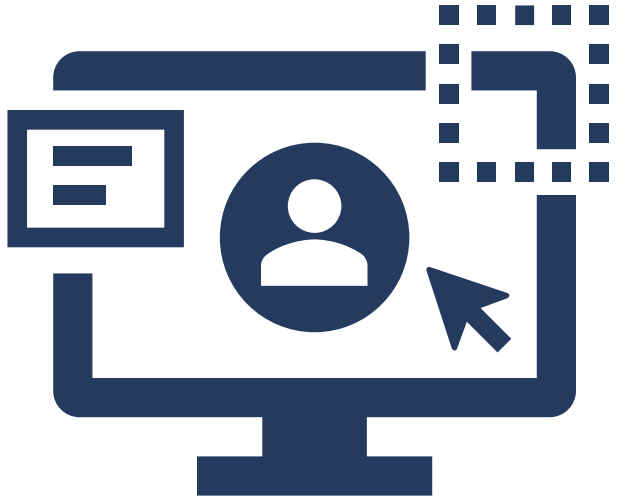
```
published_pipeline = pipeline_run.publish_pipeline(name='Batch_Prediction_Pipeline',
                                                   description='Batch pipeline',
                                                   version='1.0')
rest_endpoint = published_pipeline.endpoint
```

```
import requests

response = requests.post(rest_endpoint,
                         headers=auth_header,
                         json={"ExperimentName": "Batch_Prediction"})

run_id = response.json()["Id"]
```

Lab: Create a Batch Inference Service



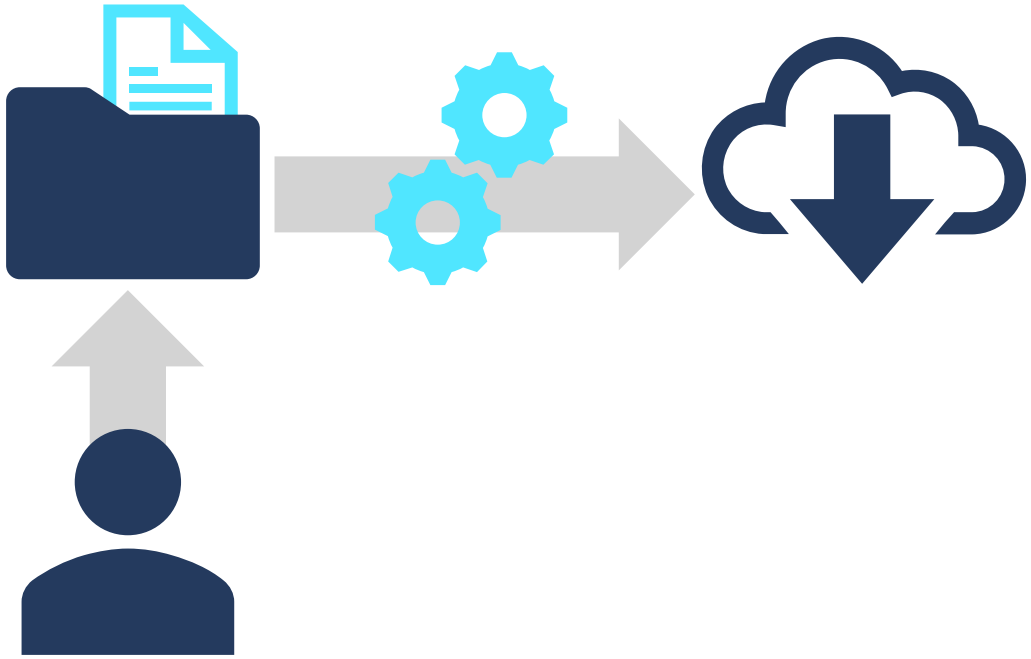
1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Create a batch inference service** exercise

Continuous Integration and Delivery

What is Continuous Integration and Delivery (CI/CD)?

A core DevOps practice for software development and deployment

- Code and other assets are managed in a central source control system
- Updates can trigger build and release processes that:
  - Apply policies to accept/reject changes
  - Integrate multiple changes into a single build
  - Perform testing and validation
  - Deploy new versions of software (including machine learning models) into staging and production environments



Azure Machine Learning and Azure Pipelines

- **Define build and release pipelines to train and deploy models**
  - Using Python or CLI
- **Install the Azure Pipelines *Machine Learning* extension:**
  - Trigger a release pipeline on model registration
  - Use predefined tasks to:
    - Run a published Azure Machine Learning pipeline
    - Profile a model
    - Deploy a model

Azure Machine Learning and GitHub Actions

- **Create a workflow to run on a specified GitHub event**

  (for example, pushing an update to a branch)

  - Use the **aml-run** action to run an Azure machine Learning pipeline or experiment
  - Use the **aml-registermodel** action to register a model
  - Use the **aml-deploy** action to deploy a model

Knowledge check

You want to deploy the model as a containerized real-time service with high scalability and token-based security. What kind of deployment target should you use?

- ☐ An Azure Container Instance (ACI)
- ☑ An Azure Kubernetes Service (AKS) inference cluster
- ☐ A multi-node compute cluster with GPUs

Which functions must the scoring script for a real-time service implement?

- ☑ *init* and *run*
- ☐ *main* and *score*
- ☐ *load* and *predict*

You want to implement a batch inference pipeline that distributes scoring on multiple nodes. Which kind of pipeline step should you use?

- ☐ PythonScriptStep
- ☐ AdlaStep
- ☑ ParallelRunStep

References

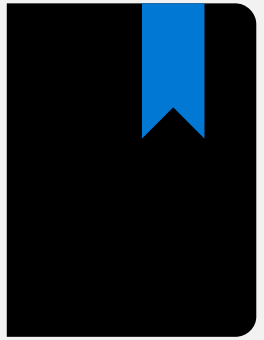
Microsoft Learn: Deploy real-time machine learning services with Azure Machine Learning
https://docs.microsoft.com/learn/modules/register-and-deploy-model-with-amls

Microsoft Learn: Deploy batch inference pipelines with Azure Machine Learning
https://docs.microsoft.com/learn/modules/deploy-batch-inference-pipelines-with-azure-machine-learning

Azure Machine Learning model deployment documentation
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where

CI/CD with Azure Pipelines documentation
https://docs.microsoft.com/azure/devops/pipelines/targets/azure-machine-learning

CI/CD with GitHub Actions documentation
https://docs.microsoft.com/azure/machine-learning/how-to-github-actions-machine-learning



Microsoft Azure