

# Introduction to pipelines

5 minutes

In Azure Machine Learning, a *pipeline* is a workflow of machine learning tasks in which each task is implemented as a *step*.

Steps can be arranged sequentially or in parallel, enabling you to build sophisticated flow logic to orchestrate machine learning operations. Each step can be run on a specific compute target, making it possible to combine different types of processing as required to achieve an overall goal.

A pipeline can be executed as a process by running the pipeline as an experiment. Each step in the pipeline runs on its allocated compute target as part of the overall experiment run.

You can publish a pipeline as a REST endpoint, enabling client applications to initiate a pipeline run. You can also define a schedule for a pipeline, and have it run automatically at periodic intervals.

## Pipeline steps

An Azure Machine Learning pipeline consists of one or more *steps* that perform tasks. There are many kinds of steps supported by Azure Machine Learning pipelines, each with its own specialized purpose and configuration options.

Common kinds of step in an Azure Machine Learning pipeline include:

- **PythonScriptStep**: Runs a specified Python script.
- **DataTransferStep**: Uses Azure Data Factory to copy data between data stores.
- **DatabricksStep**: Runs a notebook, script, or compiled JAR on a databricks cluster.
- **AdlaStep**: Runs a U-SQL job in Azure Data Lake Analytics.
- **ParallelRunStep** - Runs a Python script as a distributed task on multiple compute nodes.

**Note:** For a full list of supported step types, see [azure.pipeline.steps package documentation](#) .

To create a pipeline, you must first define each step and then create a pipeline that includes the steps. The specific configuration of each step depends on the step type. For example the following code defines two **PythonScriptStep** steps to prepare data, and then train a model.

Python

 Copy

```
from azureml.pipeline.steps import PythonScriptStep

# Step to run a Python script
step1 = PythonScriptStep(name = 'prepare data',
                        source_directory = 'scripts',
                        script_name = 'data_prep.py',
                        compute_target = 'aml-cluster')

# Step to train a model
step2 = PythonScriptStep(name = 'train model',
                        source_directory = 'scripts',
                        script_name = 'train_model.py',
                        compute_target = 'aml-cluster')
```

After defining the steps, you can assign them to a pipeline, and run it as an experiment:

Python

 Copy

```
from azureml.pipeline.core import Pipeline
from azureml.core import Experiment

# Construct the pipeline
train_pipeline = Pipeline(workspace = ws, steps = [step1, step2])

# Create an experiment and run the pipeline
experiment = Experiment(workspace = ws, name = 'training-pipeline')
pipeline_run = experiment.submit(train_pipeline)
```

## Next unit: Pass data between pipeline steps

[Continue >](#)