

Schedule pipelines

5 minutes

After you have published a pipeline, you can initiate it on demand through its REST endpoint, or you can have the pipeline run automatically based on a periodic schedule or in response to data updates.

Scheduling a pipeline for periodic intervals

To schedule a pipeline to run at periodic intervals, you must define a **ScheduleRecurrence** that determines the run frequency, and use it to create a **Schedule**.

For example, the following code schedules a daily run of a published pipeline.

Python

 Copy

```
from azureml.pipeline.core import ScheduleRecurrence, Schedule

daily = ScheduleRecurrence(frequency='Day', interval=1)
pipeline_schedule = Schedule.create(ws, name='Daily Training',
                                   description='trains model every day',
                                   pipeline_id=published_pipeline.id,
                                   experiment_name='Training_Pipeline',
                                   recurrence=daily)
```

Triggering a pipeline run on data changes

To schedule a pipeline to run whenever data changes, you must create a **Schedule** that monitors a specified path on a datastore, like this:

Python

 Copy

```
from azureml.core import Datastore
from azureml.pipeline.core import Schedule

training_datastore = Datastore(workspace=ws, name='blob_data')
pipeline_schedule = Schedule.create(ws, name='Reactive Training',
                                   description='trains model on data change',
                                   pipeline_id=published_pipeline.id,
                                   experiment_name='Training_Pipeline',
                                   datastore=training_datastore,
                                   path_on_datastore='data/training')
```

Next unit: Exercise - Create a pipeline

[Continue >](#)
