### **Steps to Build and Run the Data Pipeline in Azure DevOps**

**Step 1: Prepare the project structure**

* Create a project folder in your repository.
* Inside this folder, place your Python scripts (fetch\_data.py and process\_data.py) and a requirements.txt file.
* At the root of the repository, add the pipeline configuration file azure-pipelines.yml.

**Step 2: Write the Python scripts**

* In fetch\_data.py, add logic to fetch or simulate raw data and save it as a JSON file.
* In process\_data.py, implement code to clean, transform, or filter the raw data, then save the processed result in another JSON file.
* List the dependencies such as pandas inside requirements.txt.

**Step 3: Set up the pipeline in Azure DevOps**

1. Go to the **Azure DevOps portal**.
2. Navigate to **Pipelines → New Pipeline**.
3. Connect your GitHub or Azure Repos repository.
4. Select **Use existing YAML file** and point to the azure-pipelines.yml file.

**Step 4: Define the YAML pipeline**

The YAML file should contain instructions to:

* Trigger the pipeline on pushes to the main branch.
* Run on an ubuntu-latest agent.
* Check out the repository.
* Install Python (e.g., version 3.10).
* Install the required dependencies from requirements.txt.
* Run fetch\_data.py to generate raw data.
* Run process\_data.py to process and save the results.
* Publish the processed file as an artifact.

**Step 5: Run the pipeline**

* Save and run the pipeline.
* Monitor the logs in Azure DevOps to confirm each step executes successfully.
* At the end of the run, check the artifacts section to download the processed output.

**Outcome** By following these steps, you create an automated data pipeline. Each new code commit to the repository will trigger the pipeline, install dependencies, run the Python scripts, and produce processed outputs automatically.