Lab 5 – Monitor factory activity

ADF provides a variety of options for monitoring pipelines. ADF Studio includes a visual monitoring experience which you can use to inspect pipeline executions. You may choose to send logs to another Azure service for longer-term storage or more sophisticated analysis. This lab uses ADF Studio and explores the benefits of sending log data to Azure Log Analytics.

Lab 5.1 – Create a Log Analytics workspace

To access the Azure Log Analytics service, create a Log Analytics workspace.

1. In the Azure portal, click "Create a resource" and search for "Log Analytics Workspace". Click "Create" on the overview screen.

2. Complete the **Basics** tab like this:

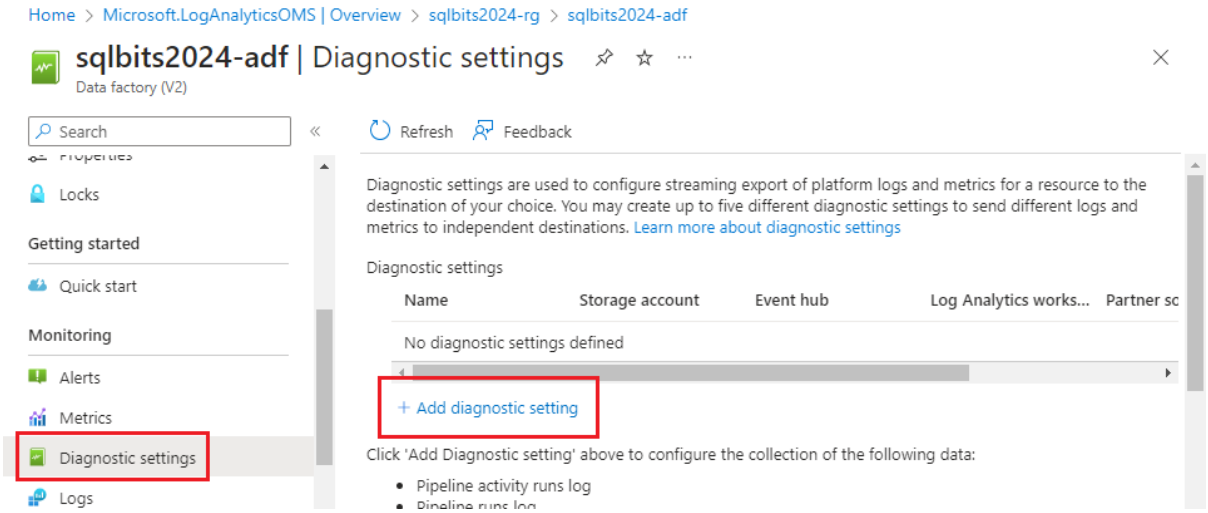
   - Choose the subscription and resource group you've been using throughout these labs.
   - Enter a name for your workspace.
   - Choose the same region you specified for your resource group.

   Click "Review + Create", then "Create".

Lab 5.2 – Configure diagnostic settings

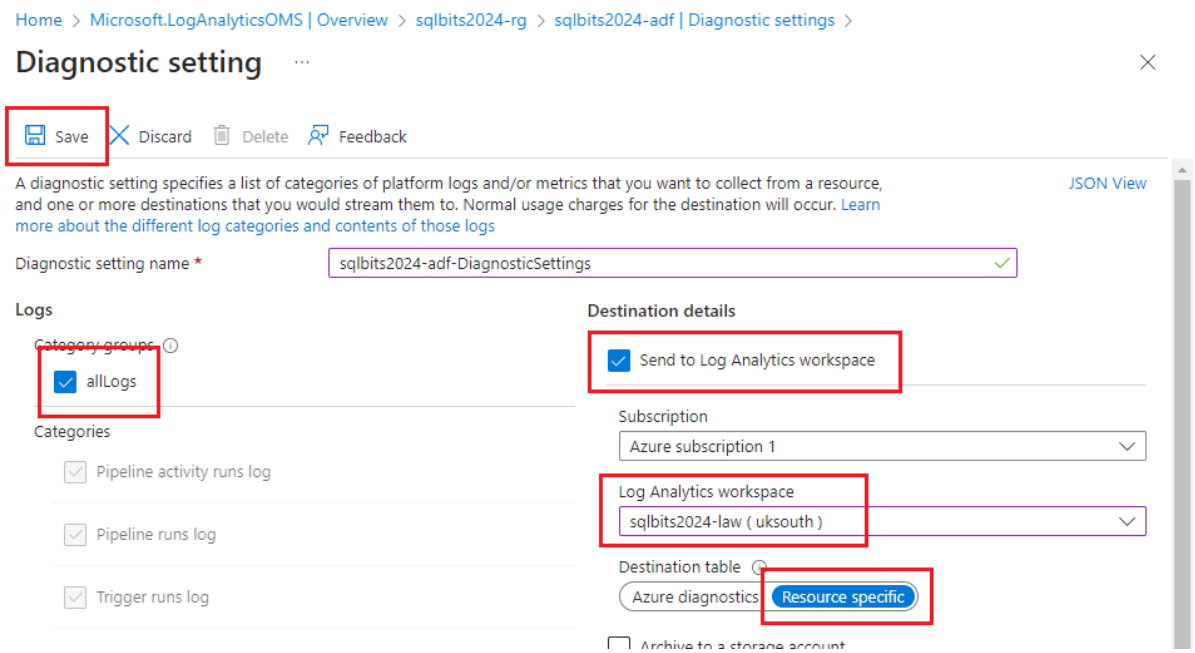
Sending log data to Log Analytics or other services from Azure Data Factory is configured in your factory's Azure portal blade.

1. On the factory's blade in the Azure portal, select "Diagnostics settings" from the sidebar (in the "Monitoring" section), then click "+ Add diagnostic setting".



2. On the "Diagnostics setting" blade, give the new diagnostic setting a name. Tick the "allLogs" and "AllMetrics" checkboxes. (The "AllMetrics" box is at the bottom of the page).

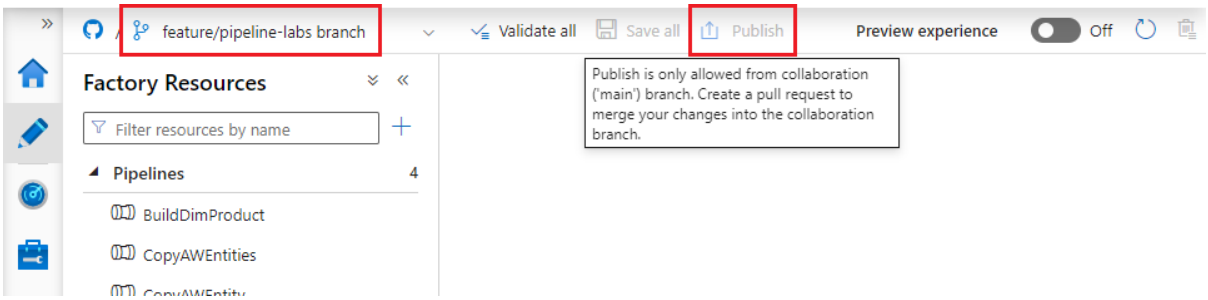
3. Tick "Send to Log Analytics workspace". Options appear to select a Log Analytics workspace – choose the one you created in Lab 5.1. Leave "Destination table" set to "Resource specific", then save your changes using the "Save" button in the top left.



Lab 5.3 – Publish factory artifacts

In Labs 1-4 you have been using ADF Studio in "develop" mode – you have been using it to create factory artifacts, save them to GitHub and run them interactively in ADF Studio. In this section you will deploy the pipelines and other artifacts created in Labs 1-4 into your ADF instance's published environment.

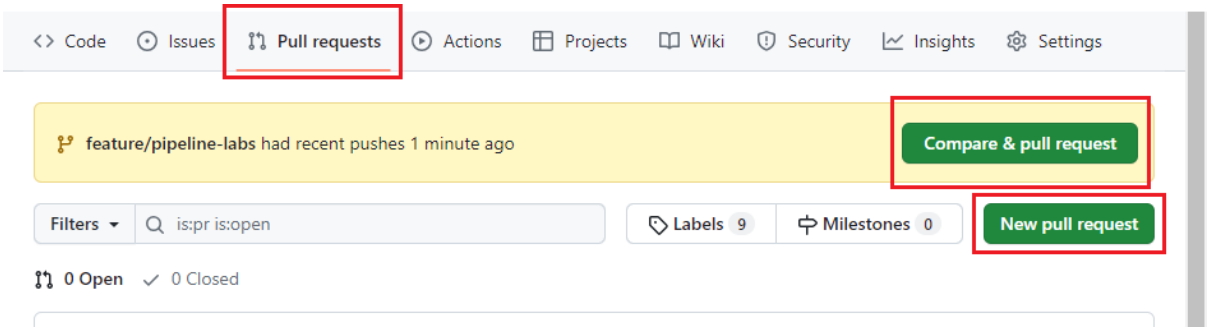
1. Open ADF Studio's Author hub, and notice the header bar's "Publish" button – this button enables development work committed to GitHub to be published into your ADF instance. If you are working in a feature branch as suggested in Lab 1.3, the button will be disabled – publish is only permitted from the linked repository's collaboration branch.



2. Use the "Validate all" button (to the left of the "Publish" button to check that all your factory resources are valid and can be published).

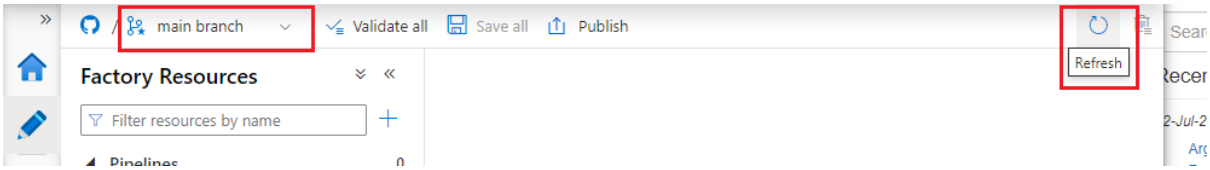
3. Use the dropdown branch selector to switch to your collaboration branch (chosen in Lab 1.3). Notice now that ADF Studio shows no pipelines, datasets or data flows – this is because you have yet to merge your feature branch into the collaboration branch.

4. To merge your feature branch, open your GitHub repository and create a pull request (PR) using the "New pull request" button on the repository's "Pull requests" tab (or by using the "Compare & pull request" button when available).



5. In a multi-engineer workflow, it is good practice to have a colleague review your PR, prior to merging into the collaboration branch. For the purposes of this exercise, consider the PR to have been approved – go ahead and merge it now.

6. After merging the branch, return to ADF Studio, and use the refresh button to reload the session from your collaboration branch. Click "Publish" to deploy the newly merged changes into the published environment, providing confirmation when required.



7. To inspect the published environment, use the dropdown branch selector to switch to live mode – this causes ADF Studio to display your published factory artifacts instead of the contents of the linked repository working branch.

Lab 5.4 – Generate log data

In previous labs you executed pipelines using the "Debug" option in ADF Studio. To generate log data, run some pipelines in the published environment. We haven't covered triggers – used to start pipelines automatically – in these labs, but you can also trigger published pipelines manually using ADF Studio.

1. Open a pipeline of your choice in ADF Studio's Author hub.

2. Click "Add trigger" above the pipeline canvas and select "Trigger now". A confirmation flyout is displayed – provide required parameter values (if applicable), then click OK to trigger the published pipeline.

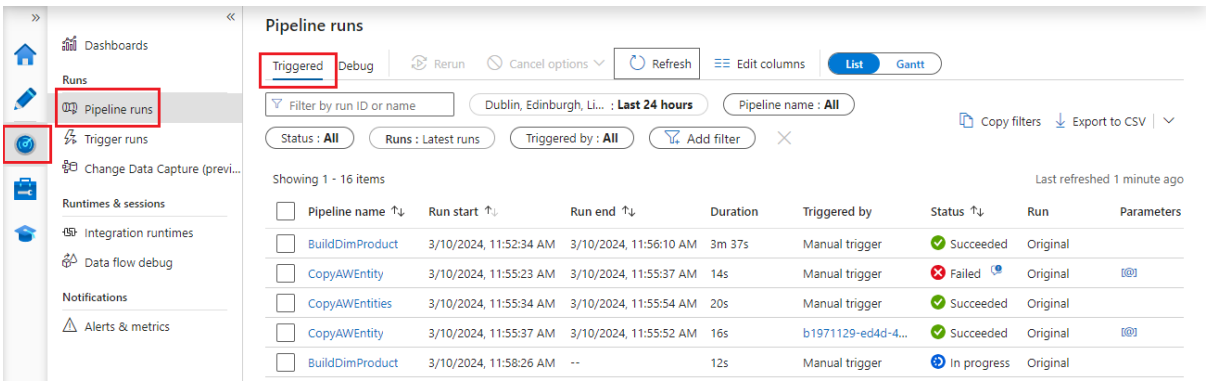
3. Repeat steps 1 and 2 a few times for different pipelines.

The purpose of running these pipelines is just to accumulate log data – in a production data factory instance, log data is generated organically from routine pipeline runs.

Lab 5.5 – Inspect logs in ADF Studio

ADF Studio's Monitoring experience allows you to inspect pipeline run history visually.

1. Open the Monitoring experience in ADF Studio by clicking the Monitor button (gauge icon). Select "Pipeline runs" from the sidebar menu then choose the "Triggered" tab to view runs of published pipeline.

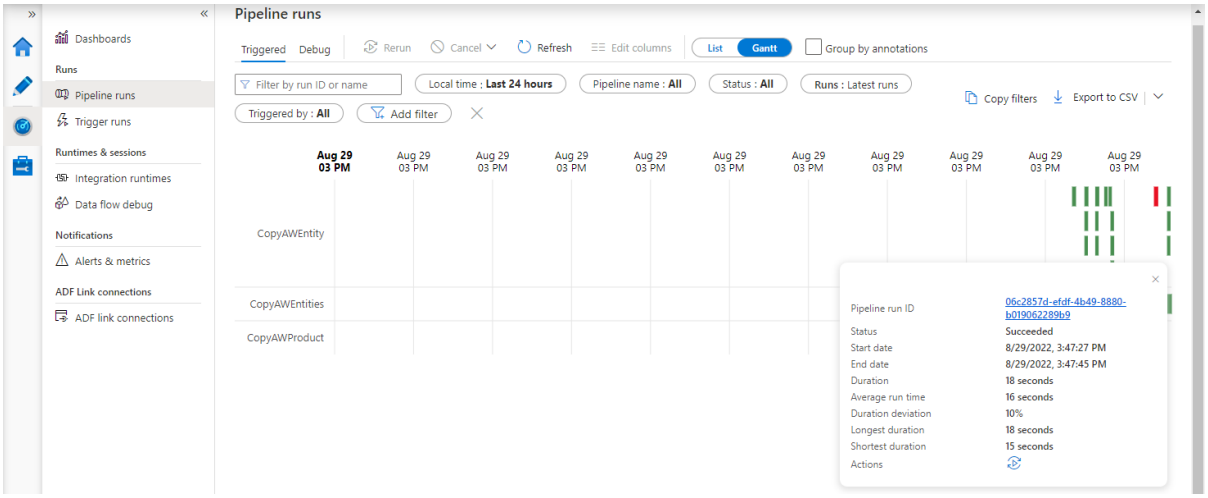


2. Explore information reported in the pipeline run list. It includes:

   - Pipeline name, run ID and start & end times.
   - Pipeline run status. For failed runs, the speech bubble icon to the right of the "Failed" status value provides failure information.
   - How the pipeline was triggered – in the screenshot, you can see pipelines triggered manually from ADF Studio, and the CopyAWEntity pipeline (from Lab 3.3) triggered by an Execute Pipeline activity run in the CopyAWEntities pipeline. Parameter values passed to each CopyAWEntity run can be inspected using the **[@]** link.

3. Hover over a pipeline name's name in the list – "Rerun" and "Consumption" buttons appear. Click on "Consumption" to view resources used by the pipeline's execution. The pipeline's name is a link – click on the link to view activity run information, presented in a similar way to the debug output you see in the authoring canvas.

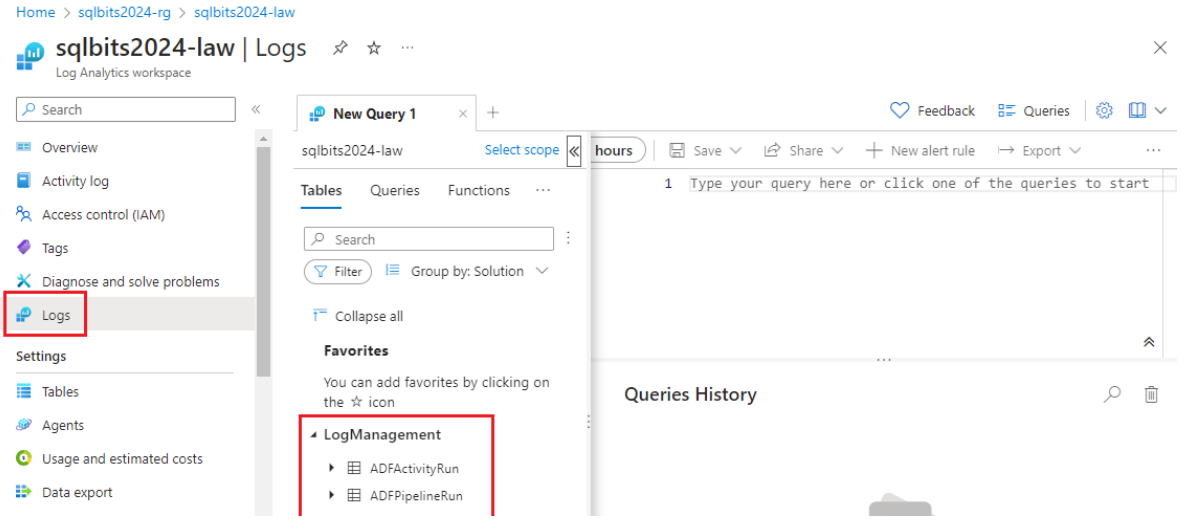
4. In the top left, move the slider from "List" to "Gantt". The Gantt chart view provides much of the same information, displaying pipeline run duration against time. Click on a bar in the Gantt chart to view detail of the pipeline's execution.



Lab 5.6 – Query Log Analytics data

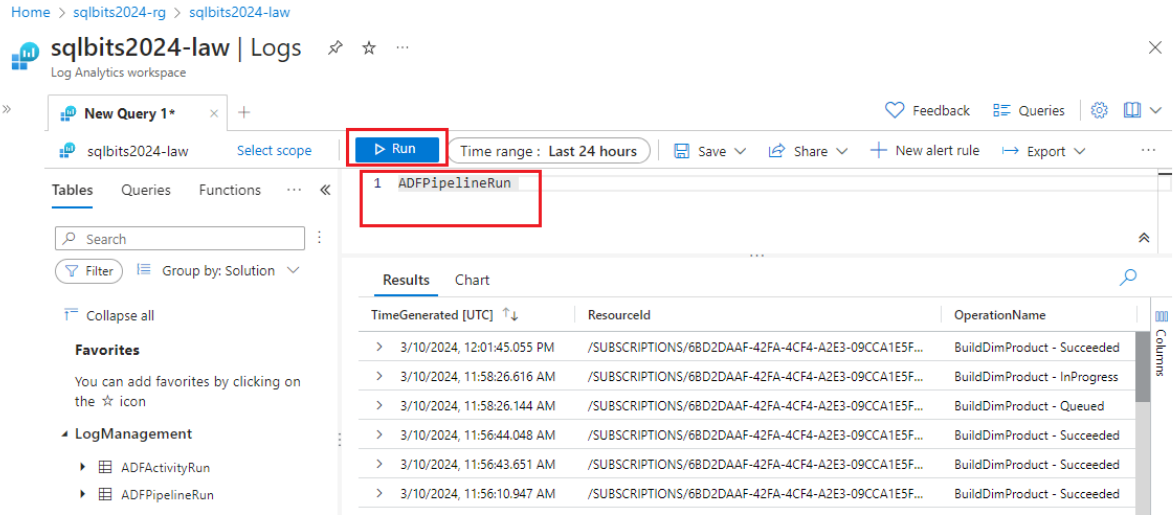
Pipeline execution data from Lab 5.4 has also been sent to Log Analytics, as configured in the diagnostic setting you created in Lab 5.2.

1. Open your Log Analytics workspace in the Azure portal and select the “Logs” item from the sidebar (in the “General” section). Close any tutorial query dialogs that appear. A query interface is displayed. Collapse the portal sidebar if you need more space.



2. The query interface includes a list of tables and a query pane. Log tables are created when the first log records are received by your Log Analytics workspace. This is not immediate – if you can’t see tables under the “LogManagement” heading, wait a few minutes and try again. Double-click on the “ADFPipelineRun” table name to add it to the query pane.

3. Log Analytics queries are written in Kusto Query Language (KQL). The table name on its own is a valid Kusto query – it means “select everything from table AdfPipelineRun”. Click “Run” to run the query.



4. This KQL query identifies the ten most recent failed pipeline runs:

```
ADFPipelineRun
  | where Status == 'Failed'
  | project PipelineName, End
  | order by End desc
  | limit 10
```

Run this or other queries to explore the logged data. The query pane provides intellisense options to help you get started quickly.

Recap

In Lab 5 you:

- created a Log Analytics workspace
- configured ADF to send log data to the Log Analytics workspace
- deployed developed factory artifacts into the published ADF environment
- generated log data from the execution of published pipelines
- interacted with pipeline logs using the ADF Studio Monitor hub and KQL queries in Log Analytics.