Lab 6 – 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 6.1 – Create a Log Analytics workspace

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

1. In the portal, click “Create a resource” and search for “log analytics”. Find the “Log Analytics Workspace” tile in the “Marketplace” search results, then use its “Create” dropdown to start configuring a new workspace.

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

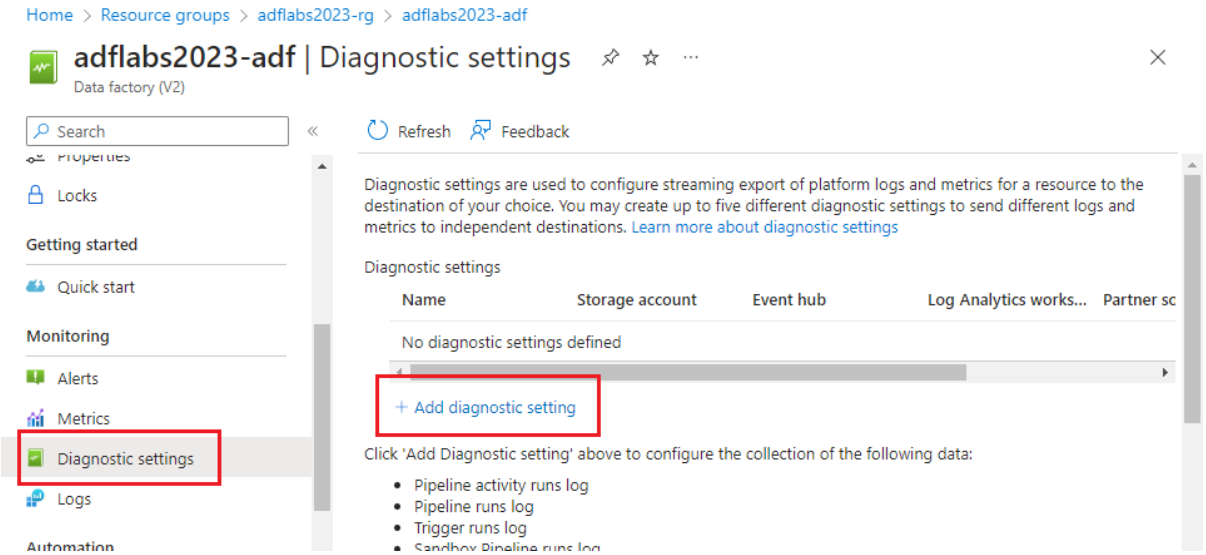
   - Choose the **subscription** and **resource group** you’ve been using throughout these labs.
   - Enter a **name** for your workspace.
   - Choose the same **region** you specified for your resource group.

   Click “Review + Create”, then “Create”. Wait for deployment to complete before proceeding to Lab 6.2.

Lab 6.2 – Configure diagnostic settings

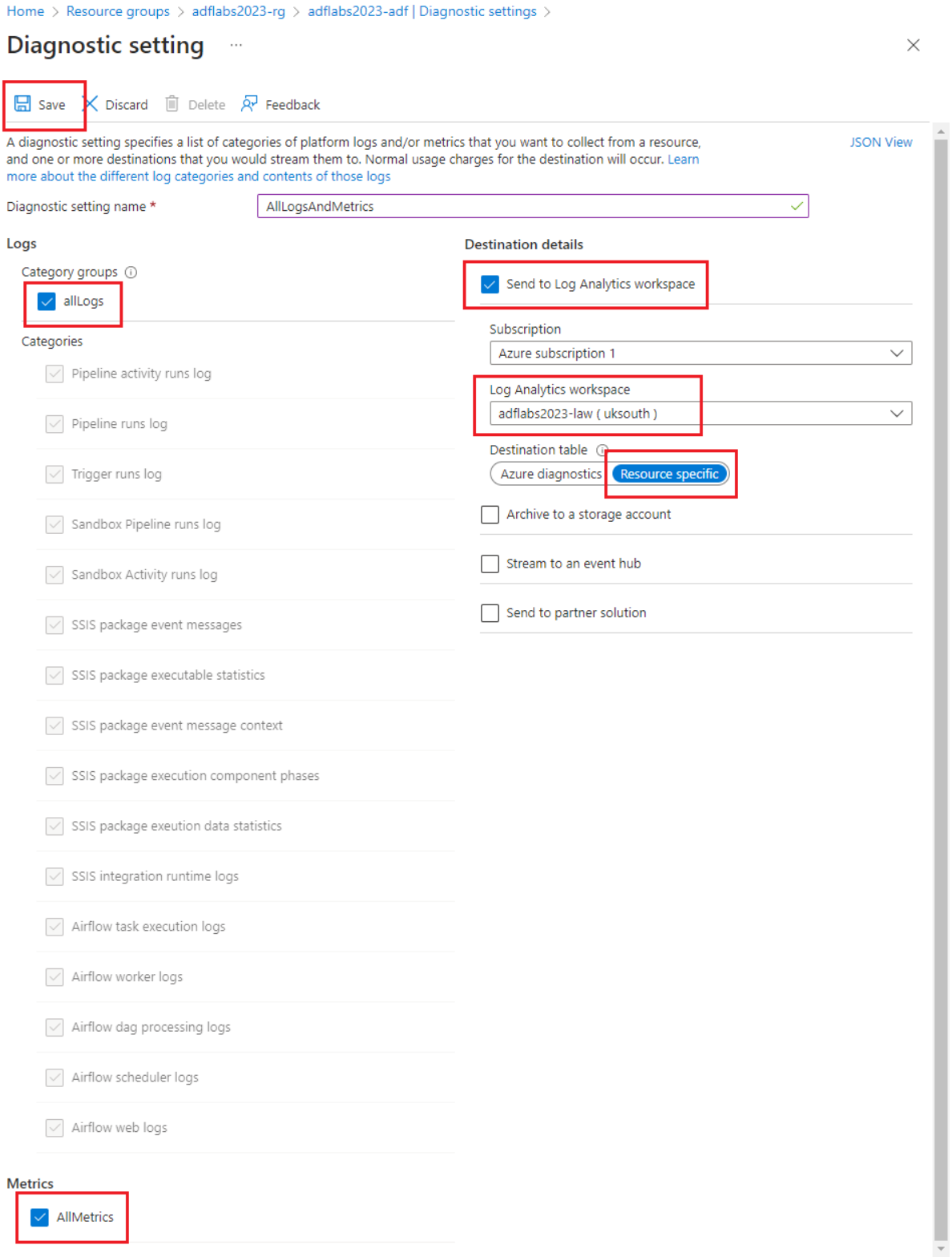
Sending log data to Log Analytics or other services from Azure Data Factory is configured in your factory’s Azure portal blade.

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.

3. Tick “Send to Log Analytics workspace”. Options appear to select a Log Analytics workspace – choose the one you created in Lab 6.1. Leave “Destination table” set to “Resource specific”, then save your changes using the “Save” button in the top left.



Lab 6.3 – 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 in these labs – a feature used to start pipelines automatically – but you can also trigger published pipelines manually using ADF Studio.

1. Open a pipeline of your choice in ADF Studio's Authoring experience.

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

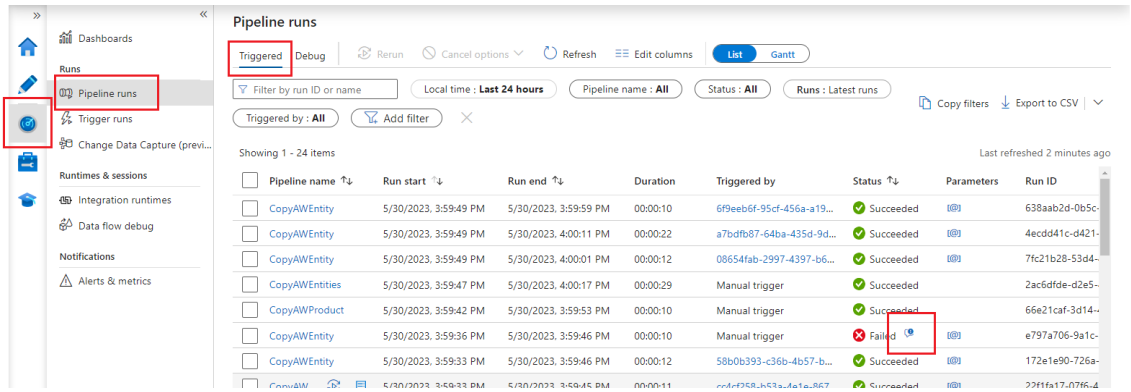
3. Repeat steps 1 and 2 a few times for different pipelines.

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

Lab 6.4 – Inspect logs in ADF Studio

ADF Studio's Monitor hub allows you to inspect pipeline run history visually.

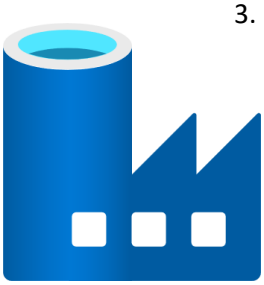
1. Open the Monitor hub in ADF Studio by clicking the Monitor button (gauge icon). Select "Pipeline runs" from the menu then choose the "Triggered" tab to view published pipeline runs.



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

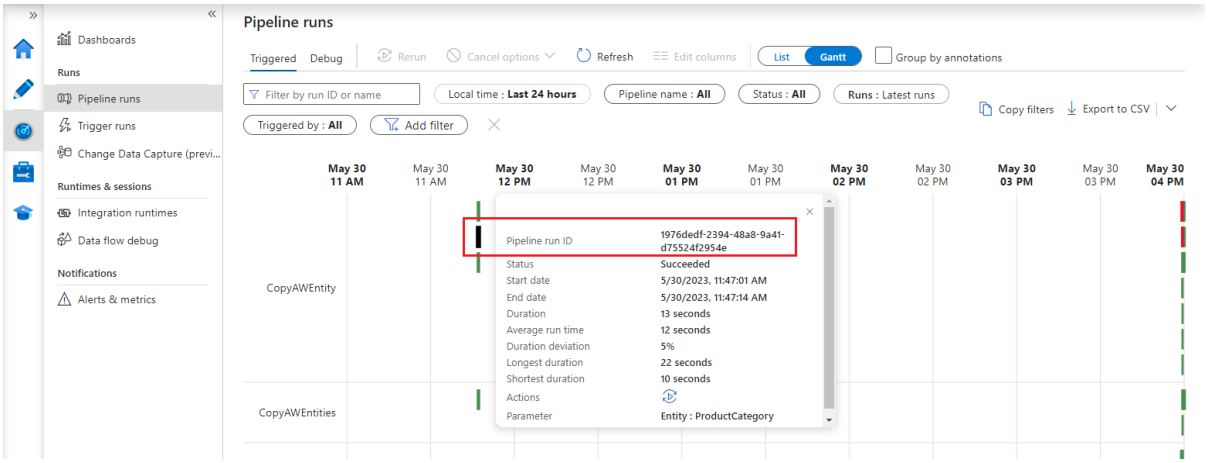
   - Pipeline name, run ID and start & end times.
   - Pipeline run status. For failed runs, the speech bubble icon to the right of the "Failed" message (indicated in the screenshot) provides failure information.
   - How the pipeline was triggered – in the screenshot, you can see pipelines triggered manually 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 in the "Parameters" column.

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.

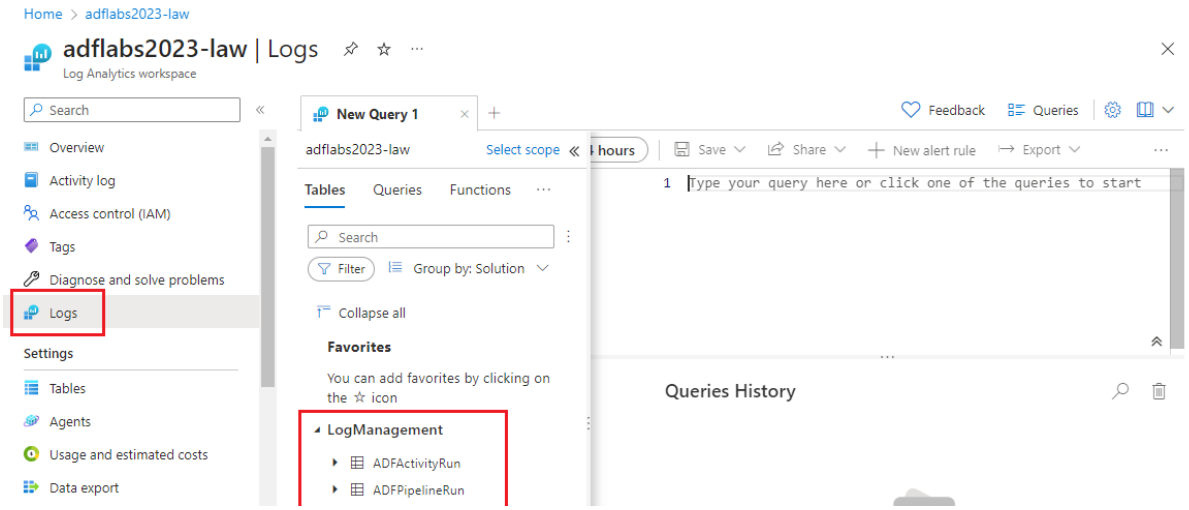


Lab 6.5 – Query Log Analytics data

Data from the pipeline executions triggered in Lab 6.3 has also been sent to Log Analytics because of the diagnostic setting you created in Lab 6.2.

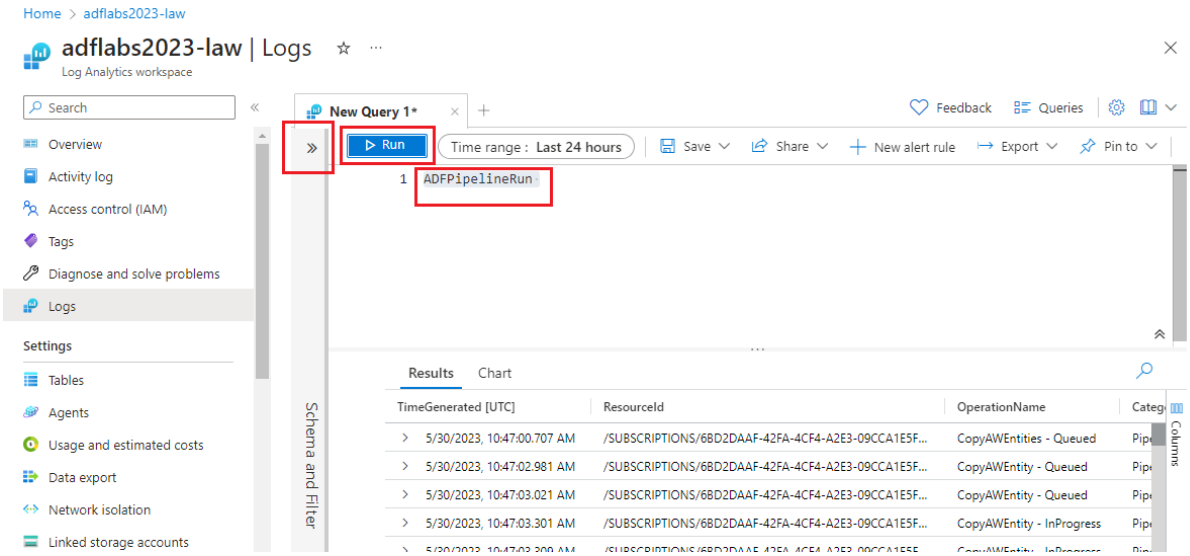
1. Open your Log Analytics workspace in the Azure portal – you can find it in the list of recent resources on the portal home page, or by selecting your resource group to see the resources it contains, or by using the search box in the portal’s top menu bar.

2. Select the “Logs” item from the sidebar (before the “Settings” section). Close any tutorial query dialogs that appear. A query interface is displayed. Collapse the portal sidebar if you need more space.



3. The “Schema and Filter” sidebar includes a list of tables and a query pane. (If not visible, expand it using the chevron icon to the top left of the query pane). In the sidebar, double-click on the “ADFPipelineRun” table name to add it to the query pane.

4. 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. (You may need to collapse the "Schema and Filter" sidebar to expose the Run button).



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

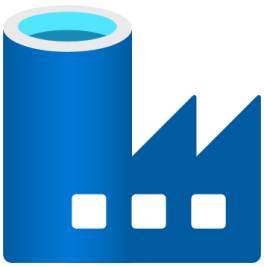
```
ADFPipelineRun
    | where Status == 'Failed'
    | project PipelineName, End
    | order by End
    | limit 10
```

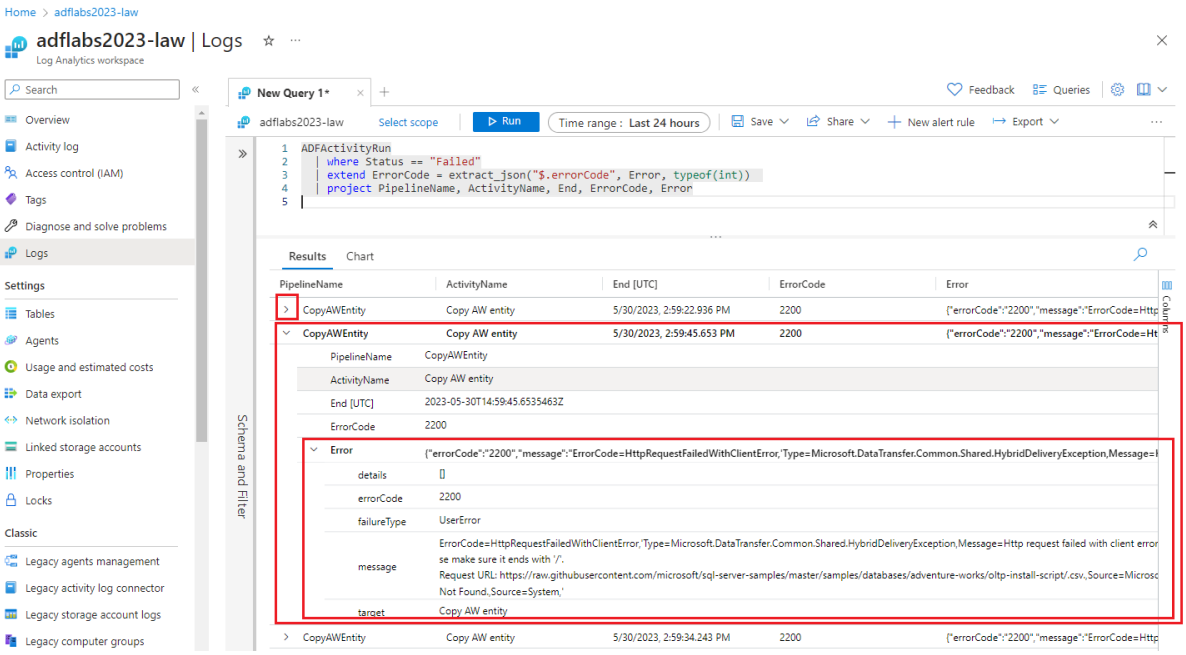
Run the query. Notice that by default KQL returns rows in **descending** order of the specified ordering attribute.

6. This query inspects failed activities and extracts their `errorCode` value:

```
ADFActivityRun
    | where Status == "Failed"
    | extend ErrorCode = extract_json("$.errorCode", Error, typeof(int))
    | project PipelineName, ActivityName, End, ErrorCode, Error
```

Run the query. Use the ">" button at the left of each result row to expand it into a list of attributes. JSON attributes can be expanded further using the nested ">" buttons:





7. Run other queries to explore the logged data. The query pane provides intellisense options to help you get started quickly. More information about Kusto can be found at https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/.

Recap

In Lab 6 you:

- created a Log Analytics workspace
- configured ADF to send log data to the Log Analytics workspace
- 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.