Pass parameters into and out of Databricks jobs in Data Factory

In this reading you can see the steps involved in the process of passing parameters into and out of Databricks jobs in Data Factory.

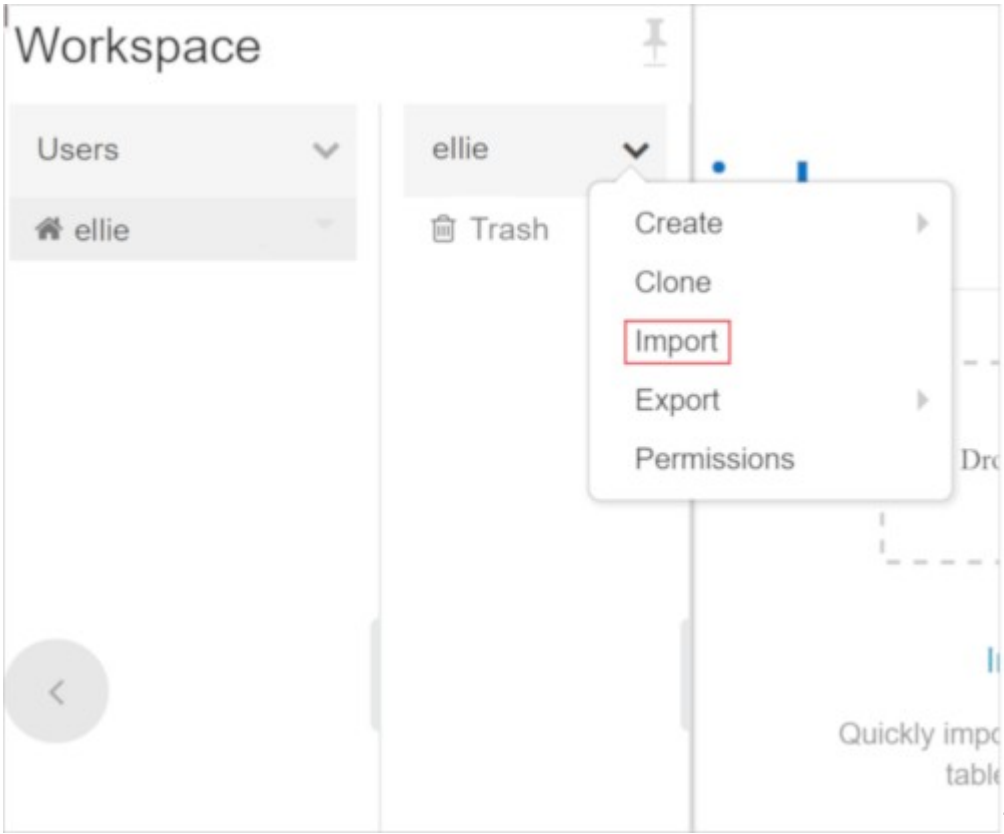
Before finishing the Notebook activity in your new Data Factory pipeline, you need to import the notebook used for this unit.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace.**

2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).

3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.

4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL:

https://github.com/solliancenet/microsoft-learning-paths-databricks-notebooks/blob/master/data-engineering/DBC/12-Production-Workloads-with-Azure-Data-Factory.dbc?raw=true

5. Select **Import**.

6. Select the **12-Production-Workloads-with-Azure-Data-Factory** folder that appears.

7. You should see the **`Record-Run`** notebook. Open the notebook to review its contents, then return to this page to continue.

Configure Databricks Notebook Activity

1. Navigate back to the Azure Data Factory pipeline.

2. With the notebook activity still selected, select the **Azure Databricks** tab near the bottom of the screen.

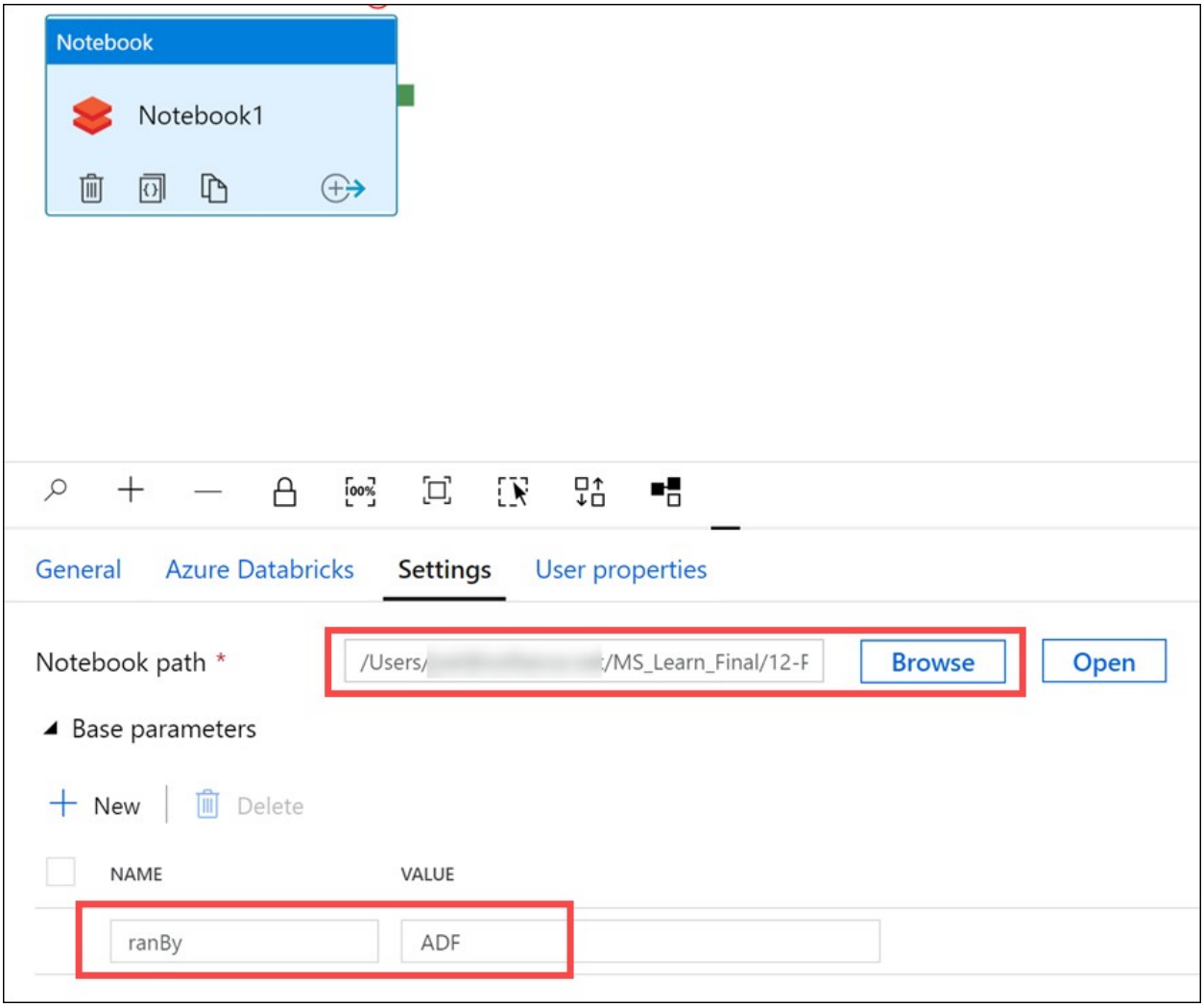
3. Select the **`Databricks linked service`** you created in the previous unit from the drop down list.

4. Under the **Settings** tab, select **Browse** to enter an interactive file explorer for the directory of your linked Databricks workspace.

5. Navigate to the **12-Production-Workloads-with-Azure-Data-Factory** directory, and pick the notebook **Record-Run**. Select **OK**.

6. Select **Base parameters** to expand a drop down, and then select **New**.

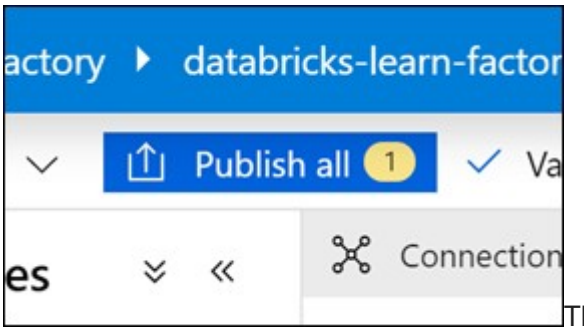
7. Under **Name** enter **`ranBy`**. For **Value**, enter **`ADF`**.



Notebook activity configuration.

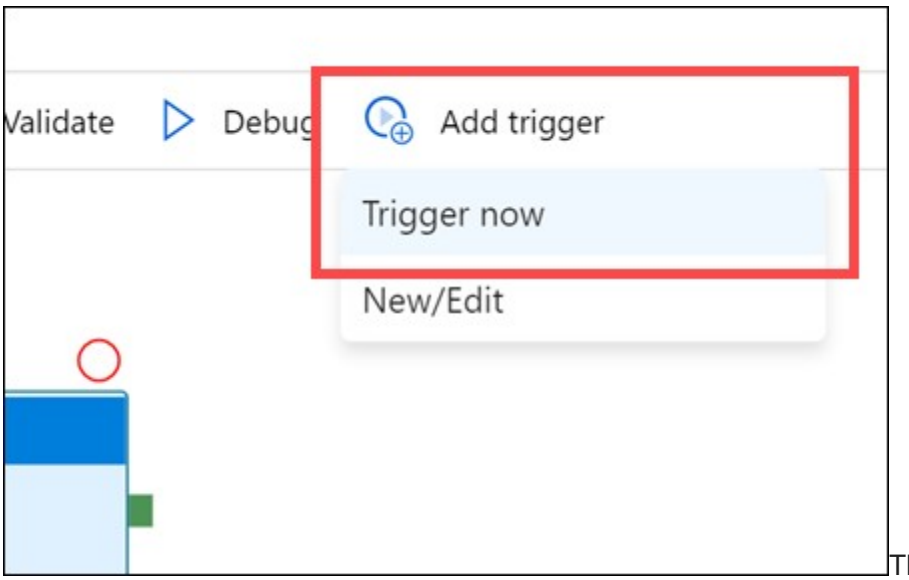
Publish and trigger the pipeline

1. At the top left, you should see a **`Publish all`** button highlighted in blue with a yellow **`1`** on it, then select **Publish**. Select this to save your configurations (this is required to trigger the pipeline).



The publish all button is shown.

2. Select **`Add trigger`** and then choose **`Trigger now`** from the drop-down. Select **`Finish`** at the bottom of the blade that appears.

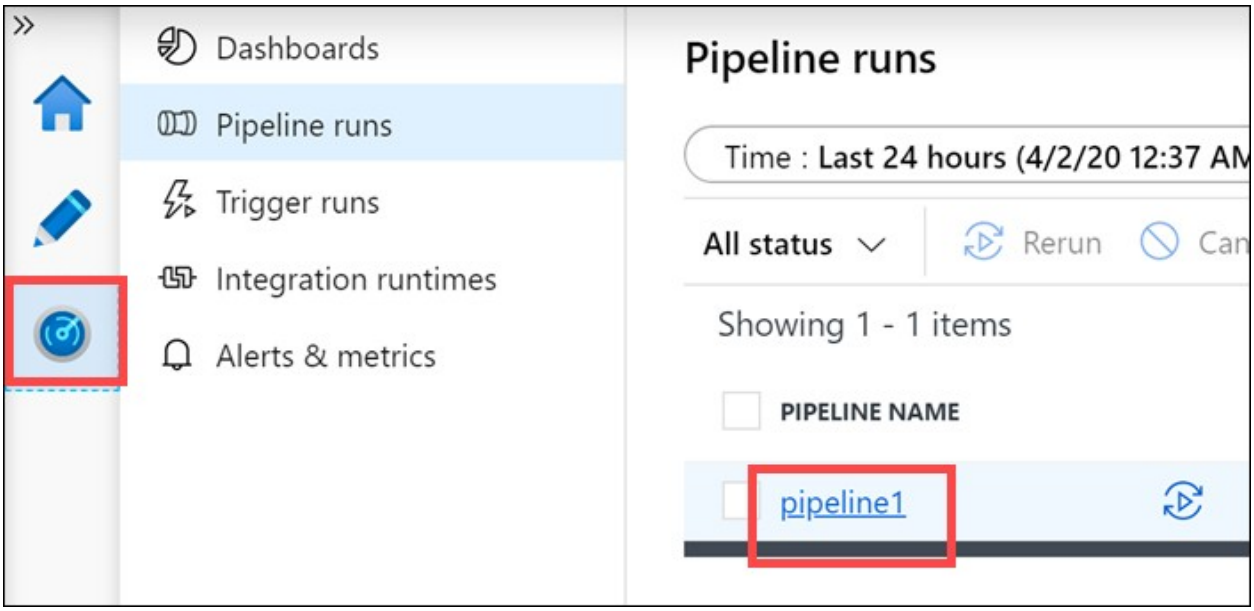


The trigger now button is shown.

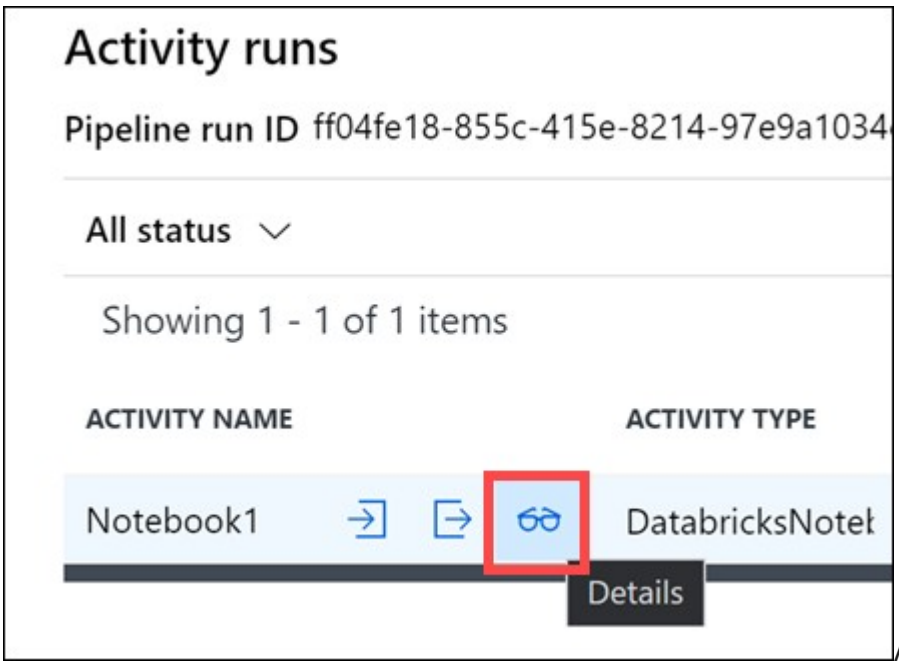
Monitor the run

1. On the left-hand menu, select the **Monitor** icon below the pencil icon. This will pull up a list of all recent pipeline runs.

2. In the **Actions** column, select the name of the action to `View activity runs`. This will allow you to see the current progress of your pipeline.

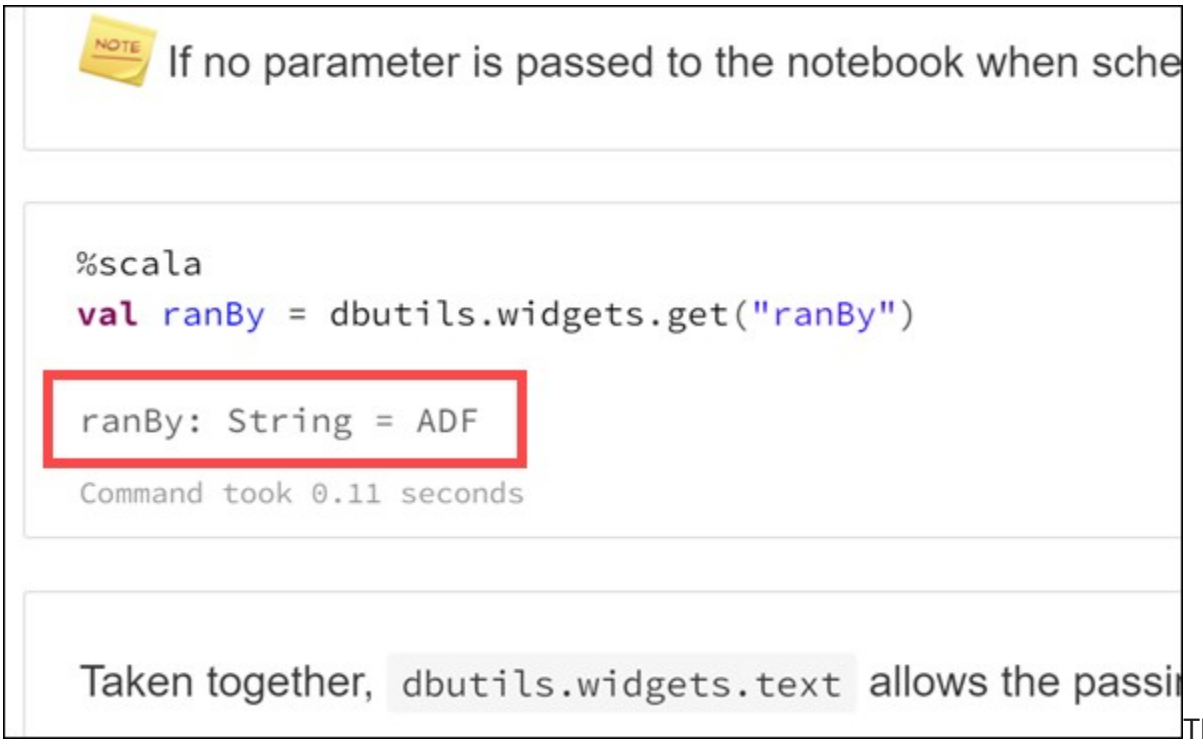


The Monitor blade is shown and an action is highlighted.
3. Your scheduled notebook will appear in the list at the bottom of the window. Select the glasses icon to view the **Details**.



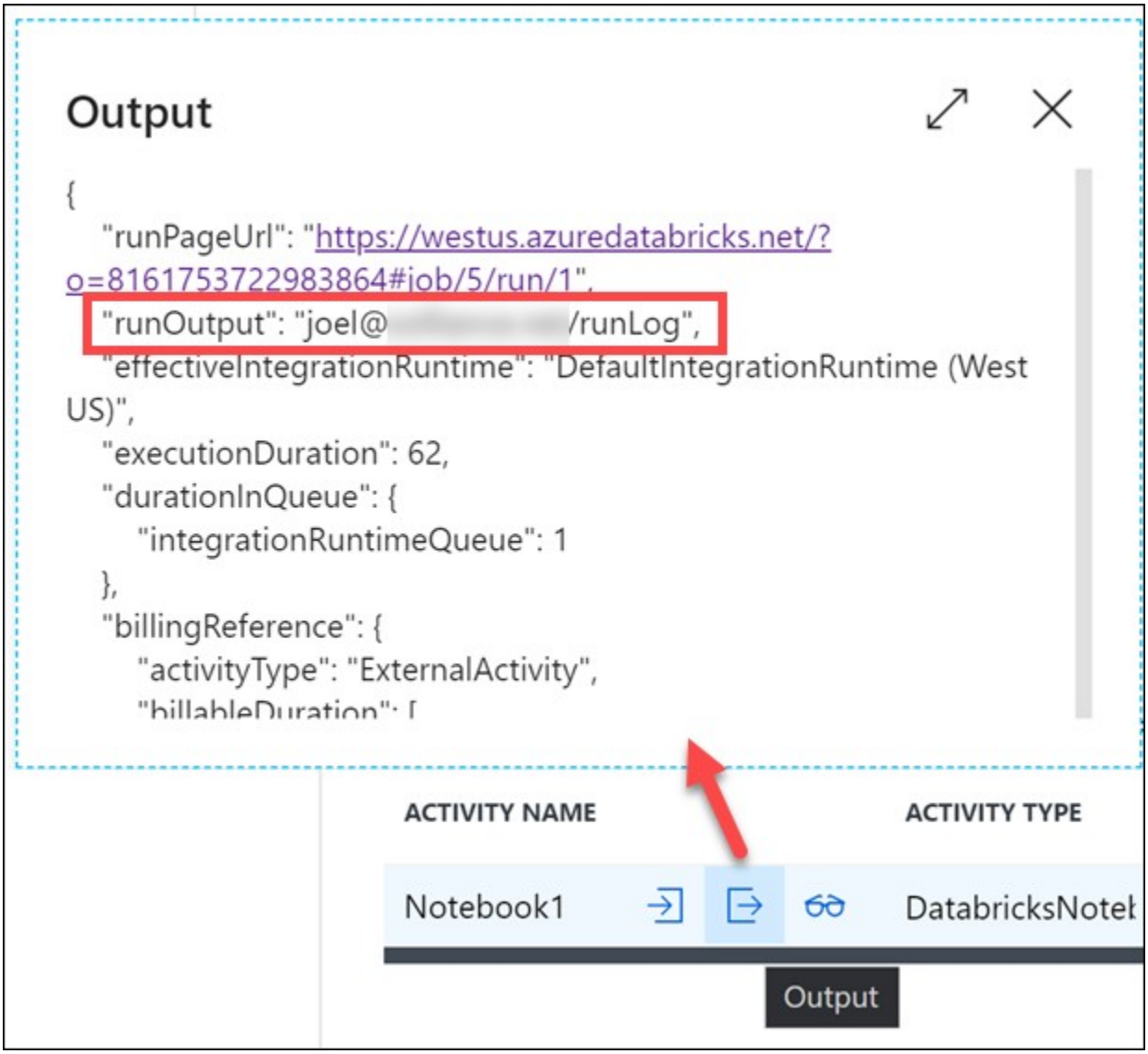
Activity details link.

4. In the window the appears, Select **Run page url**. The page that loads will be a live view of the notebook as it runs in Azure Databricks. Within the notebook, you should see the **`ranBy`** value is "ADF", which is the parameter the pipeline's notebook activity passed to the notebook on execution.



The ranBy value is displayed.

5. Once the notebook has finished running, you'll be able to view the **`Output`** of the notebook by clicking the middle icon in the **`Actions`** column. Note that the **`"runOutput"`** here is the value that was passed to **`dbutils.notebook.exit()`** in the scheduled notebook.


Output
{
"runPageUrl": "https://westus.azuredatabricks.net/?o=8161753722983864#job/5/run/1",
"runOutput": "joel@ /runLog",
"effectiveIntegrationRuntime": "DefaultIntegrationRuntime (West US)",
"executionDuration": 62,
"durationInQueue": {
"integrationRuntimeQueue": 1
},
"billingReference": {
"activityType": "ExternalActivity",
ACTIVITY NAME
ACTIVITY TYPE
Notebook1
DatabricksNotel
Output