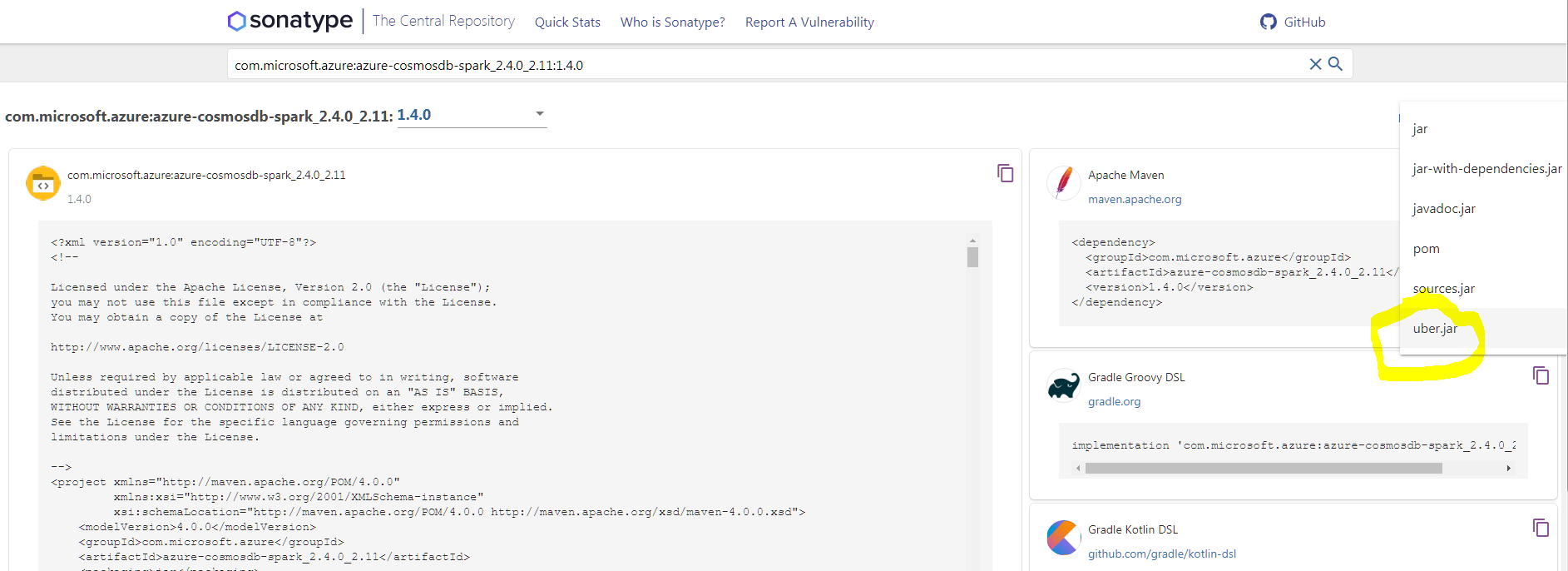
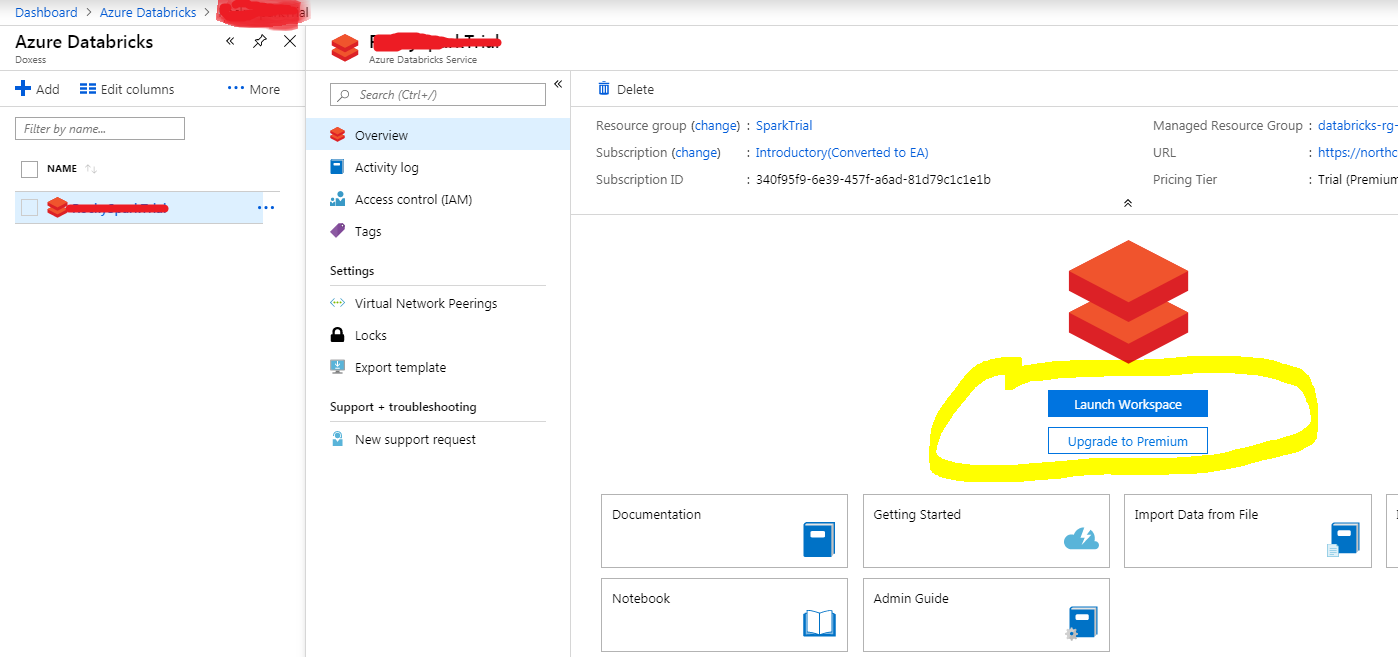
1. Dowload the latest azure-cosmosdb-spark connector from maven

<https://search.maven.org/artifact/com.microsoft.azure/azure-cosmosdb-spark_2.4.0_2.11/1.4.0/jar>

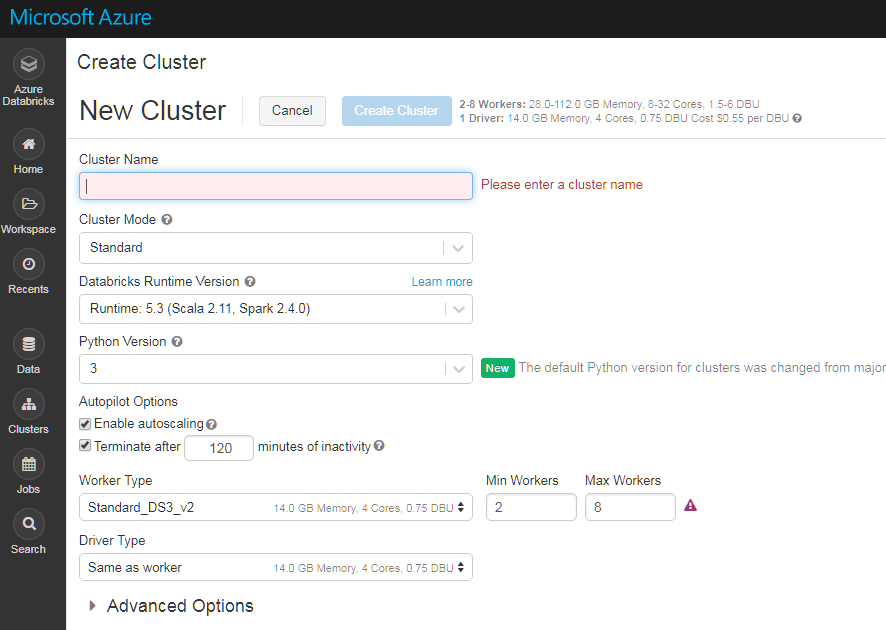
Choose uber.jar when download.



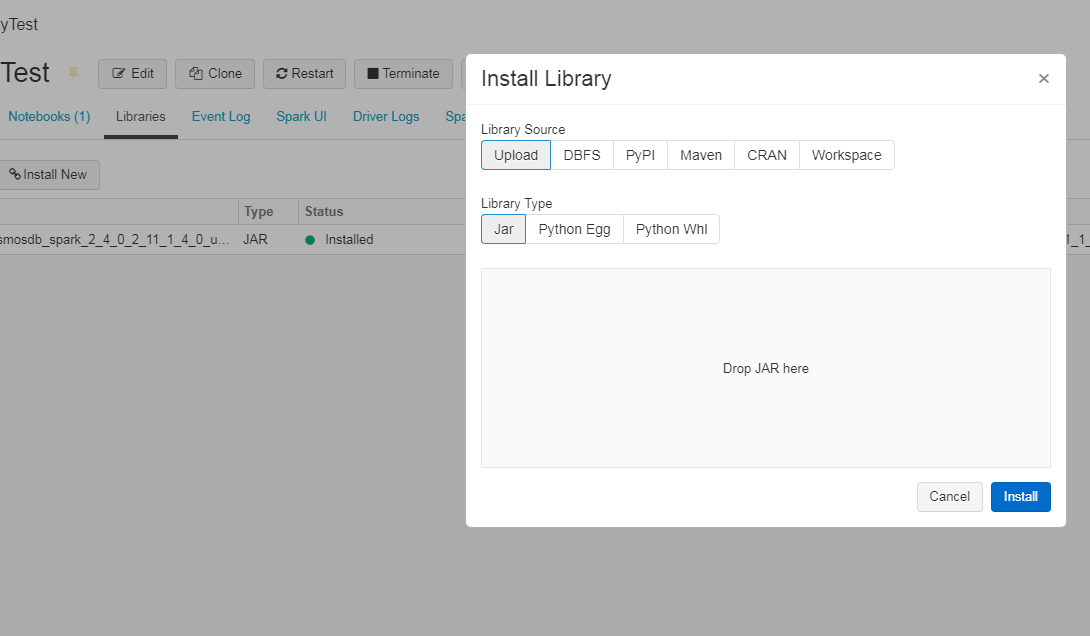
1. Create new Databricks workspace from Portal Azure and Launch Workspace



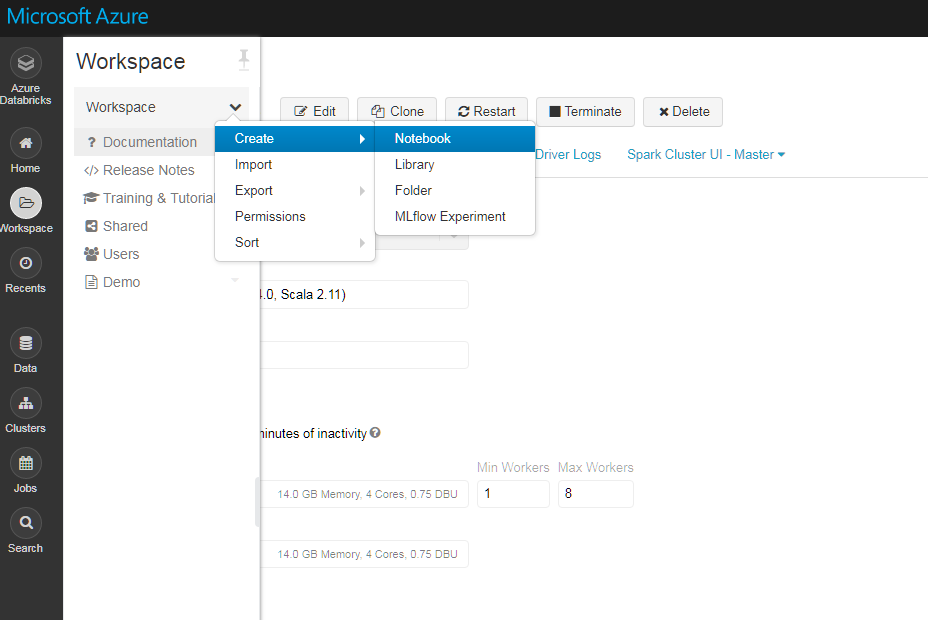
1. Create new Cluster in databricks workspace



1. Navigate to Libraries and install azure-cosmosdb-spark connector



1. Create new Notbook in Workspace



1. Attach the notebook to cluster created at step 3.
2. Write notebook and run test:
3. For how to read data from CosmosDB:

<https://github.com/Azure/azure-cosmosdb-spark>

1. For Spark programming:

<https://spark.apache.org/docs/latest/sql-programming-guide.html>

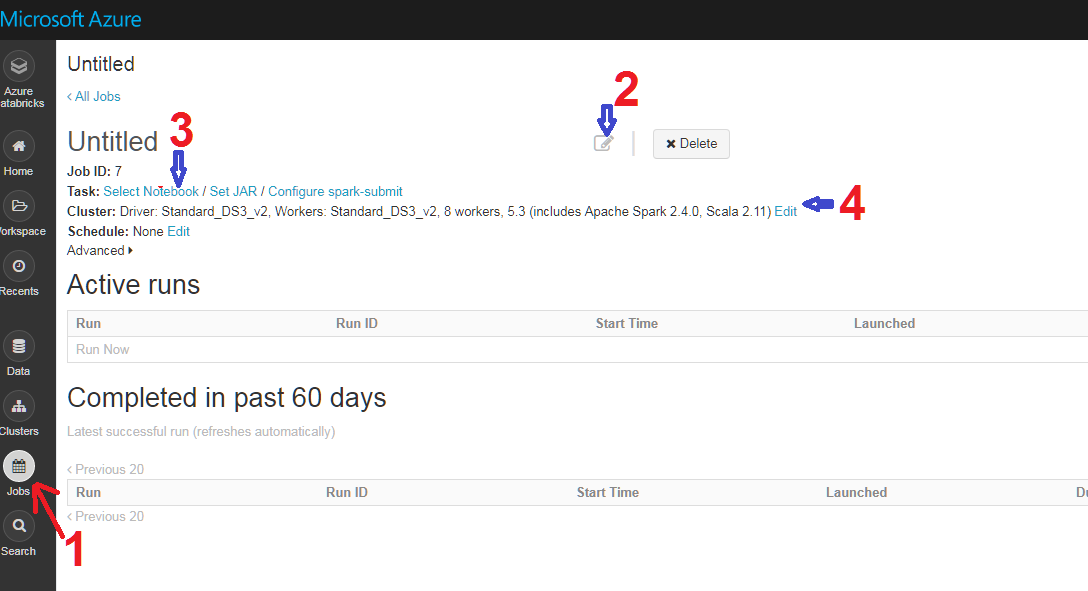
1. For Aggregations examples using Apache Spark and Azure Cosmos DB together:

<https://github.com/Azure/azure-cosmosdb-spark/wiki/Aggregations-Examples>

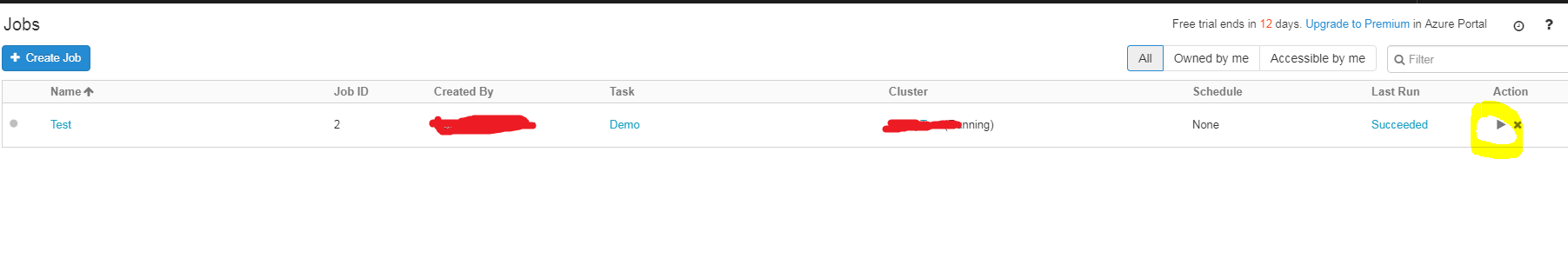
1. For how to write data from Databricks to Azure Blob

<https://gist.github.com/Zifah/ba0c3771069a11ba53969b000b038b82>

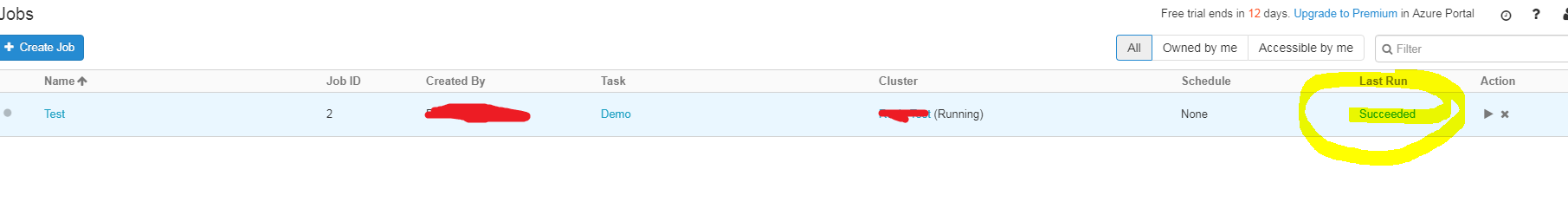
1. Create a Job on the notebook and cluster



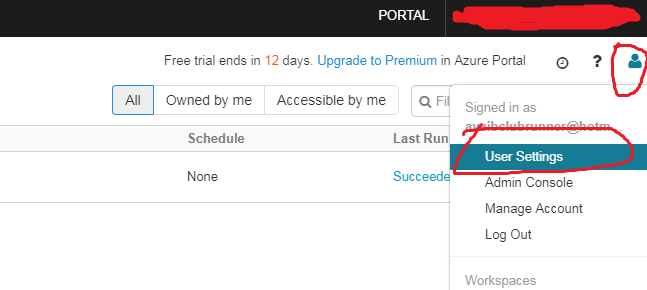
1. Run the Job to check any errors



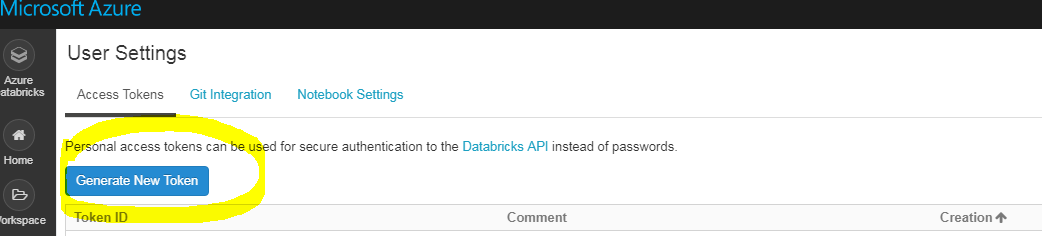
1. Click Last Run to see the details of result



1. In Databricks, go to User Settings to generate New Token for Rest API



1. Save the Token displayed in the Popup as you never see it again in Databricks

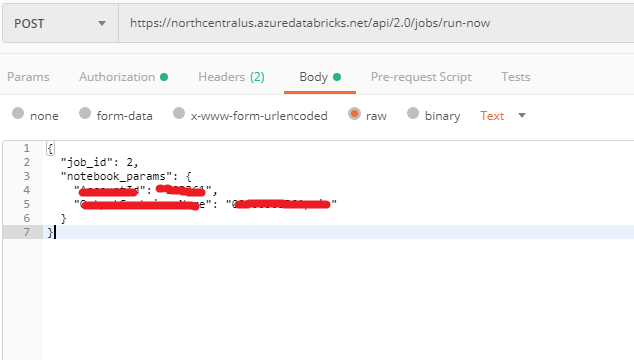


1. For details of Rest Jobs API:

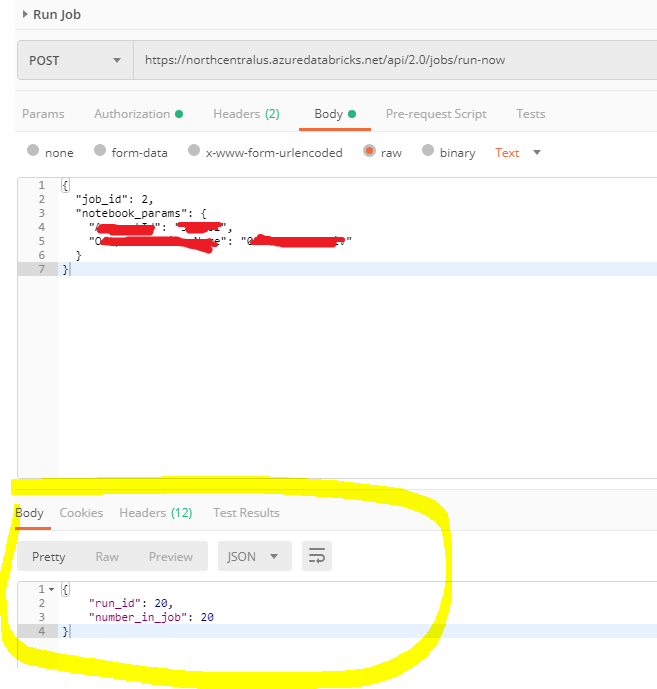
<https://docs.azuredatabricks.net/api/latest/jobs.html>

Here we only use Run Now and Runs Get Output:

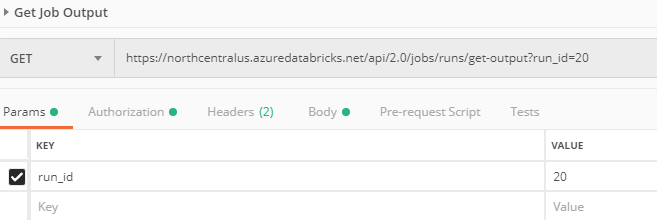
1. <https://docs.azuredatabricks.net/api/latest/jobs.html#run-now>
2. <https://docs.azuredatabricks.net/api/latest/jobs.html#runs-get-output>
3. API Examples using Postman:
4. Run the job created in step 8, which pass the job\_id and the parameters for Notebook.



It returns the run\_id if the job starts successfully.



1. Call get-output Api to get Output of the job. This is a get request with the run\_id which is returned from the above step.



The return result contains state object, which need checks. Get the “notebook\_output” when the "life\_cycle\_state": "TERMINATE” and "result\_state": "SUCCESS". The value of “notebook\_output” depends on the notebook exit.

