

**Operationalizing Analytics with Databricks & CosmosDB**

***Environment Setup Guide***

Before we can begin scoring customer profiles for propensity to buy bicycles, we need to setup a Databricks environment for training a model and performing batch scoring. We also need access to a CosmosDB collection containing customer profile data. The following steps will guide you through putting all of this in place, leaving you with the following environment.

A screenshot of a cell phone

Description automatically generated

**NOTE** Guidance in this document will be provided at a high-level, assuming you have basic familiarity with the Azure portal and the provisioning of commonly used Azure resources. Where not otherwise specified, accept default settings for various services.

**Step 1: Create a Resource Group**

1. Login to the Azure Portal
2. Create a new resource group in a region with Azure Databricks services: <https://azure.microsoft.com/en-us/global-infrastructure/services/?products=databricks>

**Step 2: Create a CosmosDB Account**

1. From the Azure Portal, create a new CosmosDB account within the resource group created in previous steps. This account should have the following characteristics:
   1. Location: same as your resource group
   2. API: *Core (SQL)*

**NOTE:** You will return to this account later to create a collection, but internal provisioning of CosmosDB resources takes longer than the Azure Portal makes it appear. Proceeding with other steps will give this time to successfully complete.

**Step 3: Create a Storage Account**

1. From the Azure Portal, create a new storage account within the resource group created in previous steps. This storage account should have the following characteristics:
   1. Location: same as your resource group
   2. Performance: *Standard*
   3. Account Kind: *StorageV2*
   4. Replication: *LRS*
2. Once created, create a container named *datasets* under **Blob Services**

**Step 4: Create a Databricks Workspace**

1. From the Azure Portal, create a new Azure Databricks deployment within the resource group created in previous steps. This deployment should have the following characteristics:
   1. Location: same as your resource group
   2. Pricing Tier: *Standard*
2. Once created, navigate to the service’s **Overview** page
3. Click on the **Launch Workspace** button to initialize your Databricks workspace.

**Step 5: Deploy a Databricks Cluster**

1. Navigate to your Azure Databricks workspace
2. Click on the **Clusters** icon in the left-hand navigation bar
3. Click on the **+ Create Cluster** button to create a new *Interactive Cluster*
4. Configure your cluster to have the following characteristics:
   1. Cluster Mode: *Standard*
   2. Python Version: *3*
   3. Enable Autoscaling: *Deselect*
   4. Workers: *4*

Accept defaults for all other cluster attributes.

1. Click on the **Create Cluster** button to launch this cluster

**NOTE** We are bouncing back and forth between Databricks and the CosmosDB environments to allow time for resources to provision between dependent steps. Feel free to re-order some of these steps if you wish to focus attention on one resource or the other.

**Step 6: Create a CosmosDB Collection**

1. From the Azure Portal, navigate to your CosmosDB account created in previous steps.
2. From the Overview page on the account, verify the **Status** shows *Online* before proceeding.
3. Click on the **+ Add Collection** item at the top of the Overview page
4. Add a collection with the following characteristics:
   1. Database ID: Select **Create New** with a database id of *app*
   2. Collection ID: *profiles*
   3. Partition Key: /id
   4. Throughput: *2000*

**Step 7: Import Lab Notebooks to your Databricks Workspace**

1. From within your Azure Databricks workspace, click on the **Workspace** icon in the left-hand navigation bar
2. Navigate to your workspace by clicking on your username and then the workspace panel to its right
3. Right-click the background of the workspace panel and select **Import**
4. On the resulting **Import Notebooks** dialog, enter the following:
   1. Import From: *URL*
   2. URL: *https://sdreadylabs.blob.core.windows.net/da-dt-ts310/DA-DT-TS310.dbc*

**Step 8: Configure Data Access Through Databricks**

1. Open the first notebook, *DA-DT-TS310/ lab\_01\_setup\_storage.py*
2. Click the drop-down in the upper left-hand corner of the page to attach the notebook to the cluster created in previous steps
3. Copy and paste your Azure Storage Account name and an associated key (accessible through the Azure Portal) into the indicated portions of the notebook script
4. Verify the cluster associated with the workbook is ready to execute scripts (as indicated by the solid green icon associated with the cluster in the upper left-hand corner of the page)
5. Run the notebook but selecting each code cell in order and clicking Shift-Enter.
6. Verify the last two cells in the notebook display data from each of the targeted files

**Step 9: Deploy Profiles to CosmosDB Collection**

1. To deploy the profiles to CosmosDB, you will first install the Azure Cosmos DB Spark connector library.
2. Open your browser to <https://docs.azuredatabricks.net/spark/latest/data-sources/azure/cosmosdb-connector.html>
3. Scroll down to the section labeled **Create and Attach Required Libraries**
4. Click on the Uber JAR file that is current (or closest to being current) with the Spark version running on your cluster deployed in previous steps.
5. Download the JAR to your PC
6. From within your Azure Databricks workspace, click on the **Workspace** icon in the left-hand navigation bar
7. Right click the background of your workspace and select **Create** and then **Library**
8. In the resulting dialog, verify the **Library** **Source** is set to *Upload* and **Library Type** is set to *Jar*.
9. Click on the **Drop Library JAR Here to Upload** box
10. In the resulting dialog, select the Uber JAR file you downloaded in previous steps
11. Click the **Open** button and give the system time to upload the JAR file
12. Once uploaded, click the **Create** button
13. In the resulting page, select the **Install automatically on all clusters** checkbox and **Confirm**
14. Wait for the **Status** for your cluster to go from *Installing* to *Installed*
15. Click on the clusters dialog in the left-hand navigation bar
16. Hover over your cluster to display its management charms
17. Click the restart charm to restart your cluster
18. Return to your workspace
19. Open the lab notebook, *DA-DT-TS310/ lab\_02\_setup\_cosmosdb.py*
20. Right-click the background of the workspace panel and select **Import**
21. Click the drop-down in the upper left-hand corner of the page to attach the notebook to the cluster created in previous steps
22. Copy and paste the URI and read-write key (accessible through the Azure Portal) for your CosmosDB account into the indicated portions of the notebook script. If you are unfamiliar with how to access these values, follow these steps:
    1. From the Azure Portal, access your CosmosDB account
    2. From the **Settings** menu in the left-hand side of the page, select **Keys**
    3. Verify you are on the **Read-write Keys** tab
    4. Locate the **URI** and **Primary** **Key** or **Secondary Key** values on this page
23. Verify the cluster associated with the workbook is ready to execute scripts (as indicated by the solid green icon associated with the cluster in the upper left-hand corner of the page)
24. Run the notebook but selecting each code cell in order and clicking Shift-Enter.
25. When the notebook has been completed, the bottom cell of the notebook will verify the count of profile documents now loaded to your CosmosDB collection. It should read 18,484.

**Step 10: Add MMLSpark library to Databricks Cluster**

The [Microsoft Machine Learning for Apache Spark](https://github.com/Azure/mmlspark/) library is used for building the bike purchase propensity model.

1. From within your Azure Databricks workspace, click on the **Workspace** icon in the left-hand navigation bar
2. Right click the background of your workspace and select **Create** and then **Library**
3. In the resulting dialog, verify the **Library** **Source** is set to *Maven*
4. In the **Coordinates** field, enter *Azure:mmlspark:0.15*
5. Click the **Create** button
6. In the resulting page, select the **Install automatically on all clusters** checkbox and **Confirm**
7. Wait for the **Status** for your cluster to go from *Installing* to *Installed*. No cluster restart is required.

**Step 10: Execute Lab Notebooks**

Open and step through the remaining notebooks in order to complete the lab:

* lab\_03\_train\_model.py – train and persist a machine learning model to predict bike purchase
* lab\_04\_batch\_score\_bulk.py – Batch score and update customer profiles with bike purchase propensity
* lab\_05\_batch\_score\_inc.py – Score incremental updates to customer profiles