**Lab: Creating an Application Solution using Azure Cosmos DB**

Lab: Creating an Application Solution using Azure Cosmos DB

*In this lab, we will build and populate a NoSQL data solution using Azure Cosmos DB. We will then take an ASP.NET MVC web application written by our development team, deploy this application to Azure and then finally connect our database to the web application.*

**Creating Database Assets**

1. Navigate to the Azure Portal ([https://portal.azure.com](https://portal.azure.com/)).
2. In the left navigation menu, click the **New** button and then select the **Databases** category.
3. In the *Databases* blade, select the **Azure Cosmos DB** option.
4. In the *Azure Cosmos DB* blade, perform the following actions:
   1. In the **ID** box, give your account a globally unique name. Name validation will run immediately to ensure that your name does not conflict with another Azure Cosmos DB account name in Azure.
   2. In the **API** section, select the **SQL** option.
   3. In the **Resource Group** section, select the **Create** new option and use the name **CosmosDB** for your Resource Group.
   4. Click the **Create** button.
5. Wait for the deployment of your Azure Cosmos DB account to complete before continuing on with this lab.
6. In the left navigation menu, click the **All Services** option.
7. In the list of services, scroll down and click the **Azure Cosmos DB** option.
8. In the *Azure Cosmos DB* blade, locate and click the account that you have just created.
9. In the *Azure Cosmos DB* blade, locate the Overview panel and record the following value:
   1. [Account Host]: Locate the **Document Endpoint** section and record the value of the endpoint. The value will have the following format: *https://[accountname].documents.azure.com:443/*
10. Locate the **Settings** section in the left menu and click the **Keys** option.
11. Click the **Read-only Keys** header to view keys that only allow you to query and read data.
12. Record the following values:
    1. [Account Key] : Locate the **PRIMARY READ-ONLY KEY** box and record the value of the key.

**Install Database Tooling and Assets**

1. Download the Compressed folder containing JSON files from this link: [docdb.json.zip](https://d37djvu3ytnwxt.cloudfront.net/assets/courseware/v1/65438080b552cda13fda2e0670cf19bf/asset-v1:Microsoft+DAT221x+1T2017+type@asset+block/docdb.json.zip)

This compressed folder includes multiple JSON files that will be imported as documents into our Azure Cosmos DB instance.

1. On your local machine, save the downloaded compressed folder.
2. Extract the contents of the compressed folder to a directory that you will be able to access later in this lab. Take a note of the directory ([JSON Files]) where you saved these files.

**Populating Database Data**

1. In the *Azure Cosmos DB* blade, locate and click the **Overview** option.
2. At the top of the blade, locate and click the **Add Collection** button.
3. In the **Add Collection** blade, perform the following actions:
   1. In the **Database Id** box, enter the name **contoso**. Take a note of this value as your [Database Name].
   2. In the **Collection Id** box, enter the name **products**. Take a note of this value as your [Collection Name].
   3. In the **Storage Capacity** section, select the **Fixed (10 GB)** option.
   4. In the **Throughput (400 - 10,000 Ru/s)** section, enter the value **400** for Request Units per second (RU/s).
   5. Click the **OK** button.
4. In the *Azure Cosmos DB* blade, click the **Data Explorer** option.
5. In the **Data Explorer** section, locate the drop-down list at the top of the page. Select the **products** collection within the the **contoso** database.
6. At the top of the blade, locate and click the **Upload** button.
7. In the **Upload Document** blade, click the folder button.
8. In the **File** dialog that appears, locate the downloaded [JSON Files]. Select all ten files, then open the files using the dialog.
9. In the **Upload Document** blade, click the **Upload** button.
10. The tool will automatically import **10** documents into your collection. You should immediately see them in the *Data Explorer* blade

**Querying Database Data**

1. In the **Data Explorer** section, locate the drop-down list at the top of the page. Select the **products** collection within the the **contoso** database.
2. At the top of the blade, locate and click the **New SQL Query** button.
3. Locate the section in the current blade where you can edit the **query text.**

For the remainder of this lab, we will refer to this section as the query editor.

1. In the *query editor*, replace the current query with the following query:

SELECT MIN(products.cost)

FROM products

1. Click the **Execute Query** button.
2. In the *Results* blade, observe the results of your query.
3. In the *query editor*, replace the current query with the following query:

SELECT VALUE MIN(products.cost)

FROM products

1. Click the **Execute Query** button.
2. In the *Results* blade, observe the results of your query.
3. Take a note of the single value returned in the result array. This is the least expensive entity in the database..
4. In the *query editor*, replace the current query with the following query:

SELECT VALUE

{

guid: products.id,

name: products.name,

financial: {

"value": products.cost

},

specification: {

vendor: products.vendorid,

style: products.color

}

}

FROM products

WHERE products.vendorid = 28

1. Click the **Execute Query** button.
2. In the *Results* blade, observe the results of your query.
3. Take a note of the single document returned in the result array. This complex document represents the only match in our database for Vendor /#28.

**Deploy and Test Web Application**

1. Click the following button to go to the **Template Deployment** blade in the Azure portal:

[[Deploy to Azure](https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure-Architecture-Workshop%2Fasset-tracking%2Fdocumentdb%2Fdeploy%2Farmdeploy.json)](https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure-Architecture-Workshop%2Fasset-tracking%2Fdocumentdb%2Fdeploy%2Farmdeploy.json)*(This link will open the Azure Portal. To continue reading the lab instructions, you should open this link in a new browser window or tab.)*

This button will redirect you to the Azure Portal and pre-populate an application template automatically for you. The application template contains a web application and code to connect to your database.

1. Once you reach the Azure Portal, you will immediately see the *Custom deployment* blade. In the *Custom deployment* blade, perform the following actions:
   1. In the **Resource Group** section, select the **Use existing** option and then select the **CosmosDB** Resource Group from the list.
   2. In the **Terms and Conditions** section, read the Marketplace terms and then click the **I agree to the terms and conditions stated above**checkbox.
   3. Click the **Pin to dashboard** checkbox to ensure that the Resource Group is visible on your dashboard and that the Resource Group is opened automatically when the deployment is complete.
   4. Click the **Purchase** button.
2. Wait for the Resource deployment operation to complete before continuing on with this lab.

This template will deploy an App Service Plan instance and a Web App instance to your Resource Group. These instances will have automatically generated names to ensure they have unique URLs. You will validate that the Web App can connect to your database.

1. Once the deployment has completed, you will immediately see the *Resource Group* blade. In this blade, locate and select your newly created **Web App** instance.
2. In the *App Service* blade, locate the **Settings** section in the left menu and click the **Application Settings** option.
3. In the *Application Settings* blade, scroll down and locate the **App Settings** section. Observe the settings that have been created in your Web App.
4. Locate and click the **Overview** option in the left menu.
5. In the *App Service* blade, locate and click the **Browse** button at the top of the blade. This will open your web application in a new browser window or tab.

You will immediately see a warning indicating that you need to fill in your configuration settings before the dashboard will work.

1. On the **Asset Tracking System**, click the **Configure** link.
2. On the configuration page, perform the following actions:
   1. In the **Host URL** box, enter the [Account Host] value recorded earlier in this lab.
   2. In the **Read-Only Key** box, enter the [Account Key] value recorded earlier in this lab.
   3. In the **Database Name** box, enter the [Database Name] value recorded earlier in this lab. This value should be **contoso**.
   4. In the **Collection Name** box, enter the [Collection Name] value recorded earlier in this lab. This value should be **products**.
   5. Click the **Save Configuration** button.
3. Click the **Dashboard** link.
4. Observe the tables and headers visualized in the application.
5. Take a note of the **Average Cost** of assets in your database.
6. Take a note of the **Total Assets** count in your database.

Source: edx.org

Microsoft: DAT221x**Introduction to NoSQL Data Solutions**