# Azure Discovery Days 2019

## Data Analytics & Near Real Time Intelligence with Azure - Hands-On Lab Guide

## Lab 1: Ingest and Store

### Summary

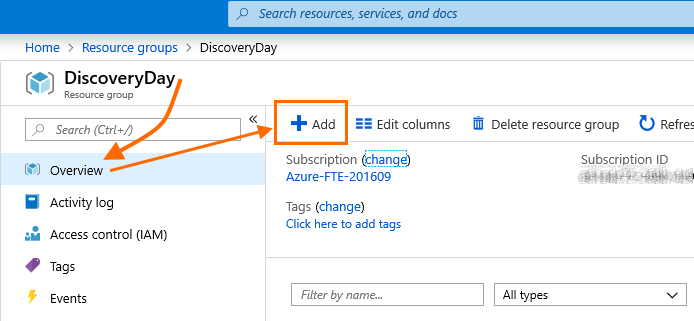
In this hands-on lab, you will:

1. Set up an Azure storage account and a blob container
2. Ingest source file-based data to your Azure storage account
3. Deploy an Azure Databricks cluster
4. On Databricks, run a Jupyter notebook to load, transform, and emit data for use in the next lab.

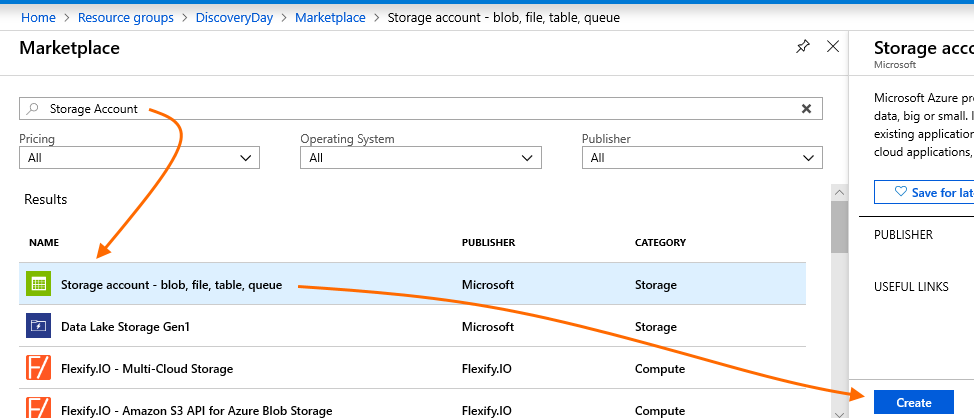
### Task 1 – Set up an Azure storage account and a blob container

First you will create an Azure storage account in your resource group. This is where you will copy the source data files we will use later in this lab.

Start in the Azure portal, in the Resource Group you created in lab 0. Ensure you are on the “Overview” blade.

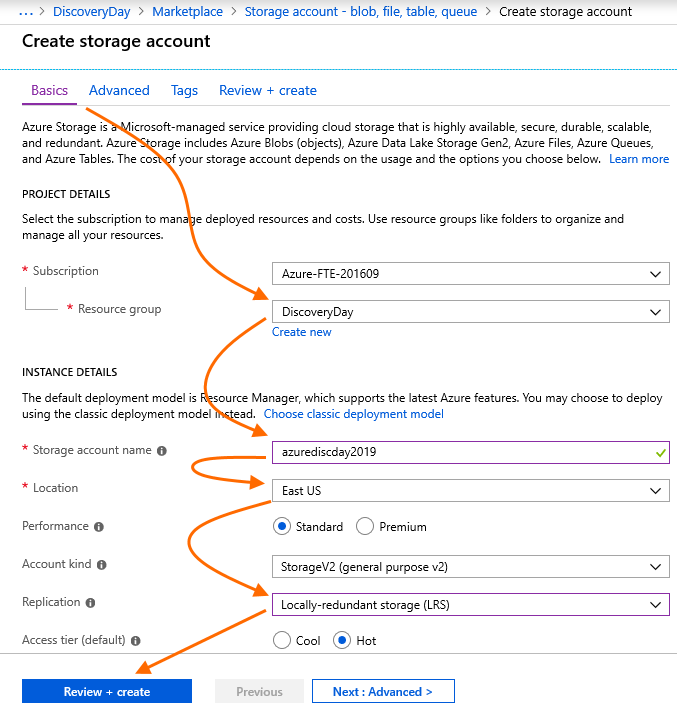


In the Search box, type “Storage Account” and hit Enter. From the results, click on “Storage account – blob, file, table, queue”. On the next blade, click “Create”.

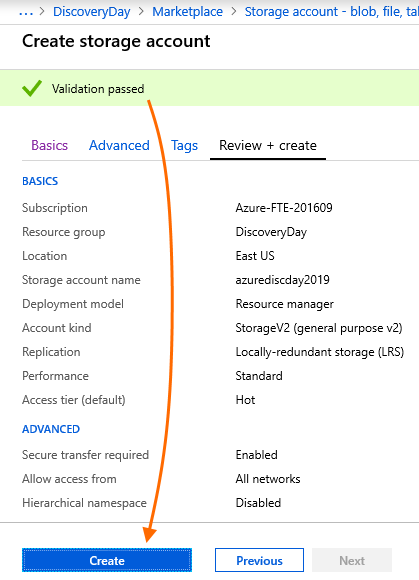


The create flow is divided into several tabs: “Basics”, “Advanced”, “Tags”, and “Review + create”. For this lab, we will only use “Basics” and “Review + create”, but please feel free to explore the other tabs.

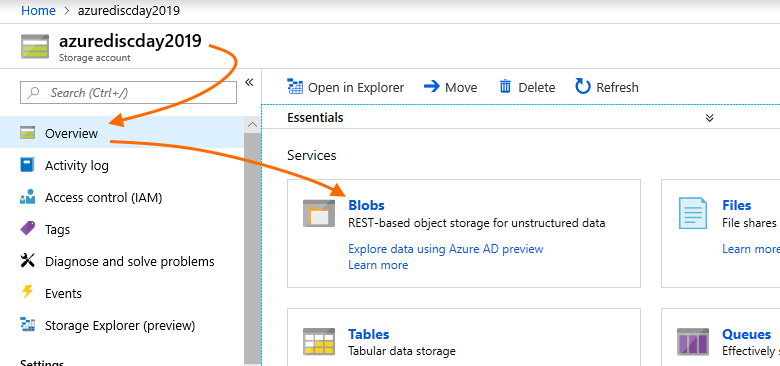
On the “Basics” tab, ensure your Resource Group is selected. Then provide a storage account name; this must be globally unique. Set the Azure region (Location) and set Replication to “Locally-redundant storage (LRS)” (this does not provide cross-region DR but is enough for this lab). Leave other settings at their defaults. Then click “Review + create”.



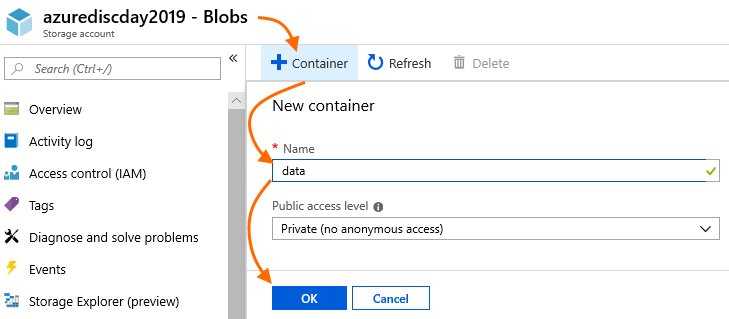
On the validation screen, click “Create”.



Wait for the deployment to complete (see Notifications, as shown in lab 0). Then go to your new storage account’s Overview blade. Under Services, click “Blobs”.



On the Blobs page, click “+ Container” and create a new blob container. The screenshot shows a container name of “data”, but (subject to naming rules) please feel free to use a different name; however, be sure to adjust accordingly in later tasks and labs. Leave the public access level at its most secure default, “Private (no anonymous access)”. Then click “OK”.



After the container is successfully created and you can see it on the blob container page, this task is complete.

NOTE: for later tasks and labs, note the following information (e.g. in a OneNote page):

* Storage Account name
* Storage account key
* Blob container name

You can retrieve the key from the storage account’s “Access keys” blade. You can use either the primary or secondary key.

### Task 2 – Ingest source file-based data to your Azure storage account