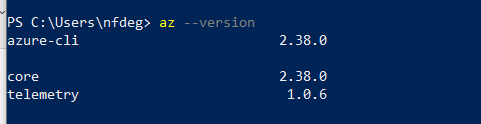
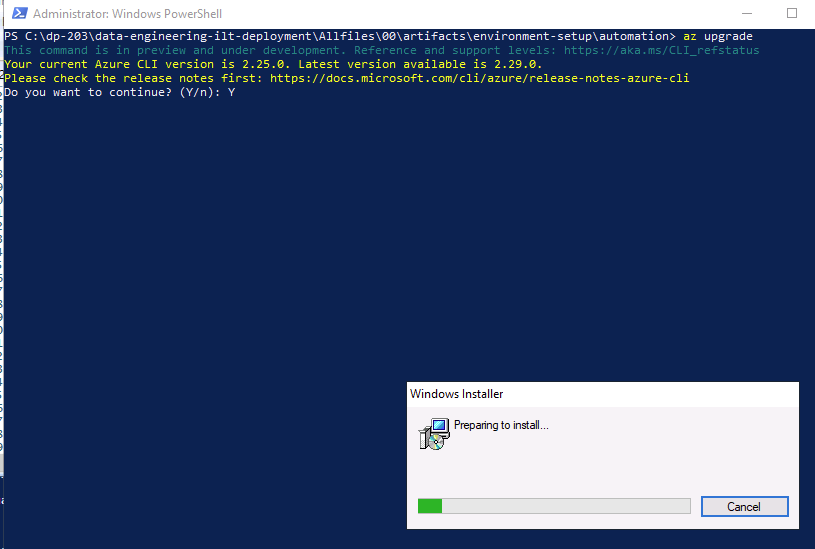
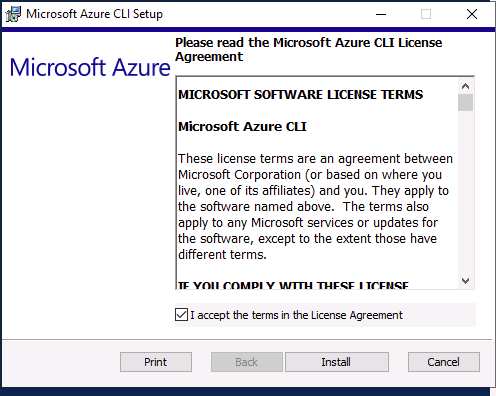
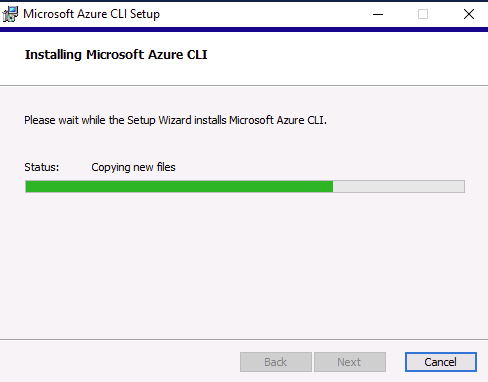
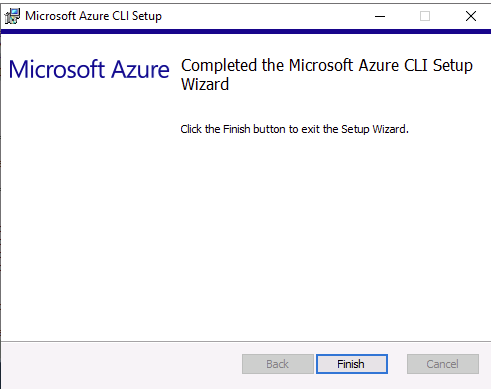
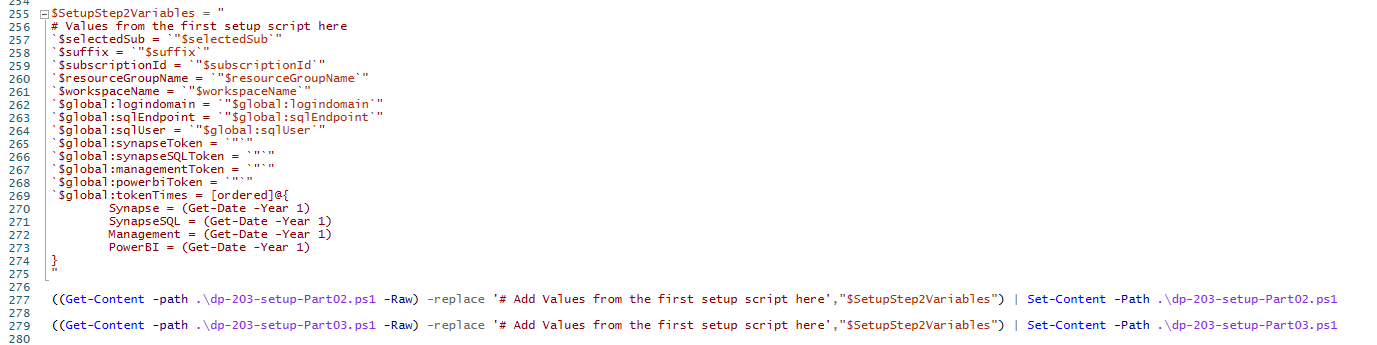
Lab 1 Notes

1. Before executing C:\dp-203\data-engineering-ilt-deployment\Allfiles\00\artifacts\environment-setup\automation\dp-203-setup-Part01.ps1 make the following changes:  
   line 110: replace “australiaeast” by “eastus”, “westeurope” by “eastus2”. One student had issues with “uksouth”, replace by “centralus”  
     
   We get an issue with line 58 if we are using an older version of the CLI.   
   
2. If it is less than 2.37 you can update with **az upgrade,** otherwise on line 58 we have to use objectId instead of id. 🡺 The VM was update to 2.38, but then it was updated again to remain using an older version. In this case use objectId. The GitHub site has been switching between id and objectId. At this moment it is back objectId. We need to always check which one is currently being used and match with the version of AZ CLI in the VM.  
     



1. If we need to run the first script again, we have to manually change the other 2 so that the variables are inserted properly:



So, you need to edit the 2 scripts and make sure that # Add Values from the first setup script here is added back. After executing the first script, the second one will have:

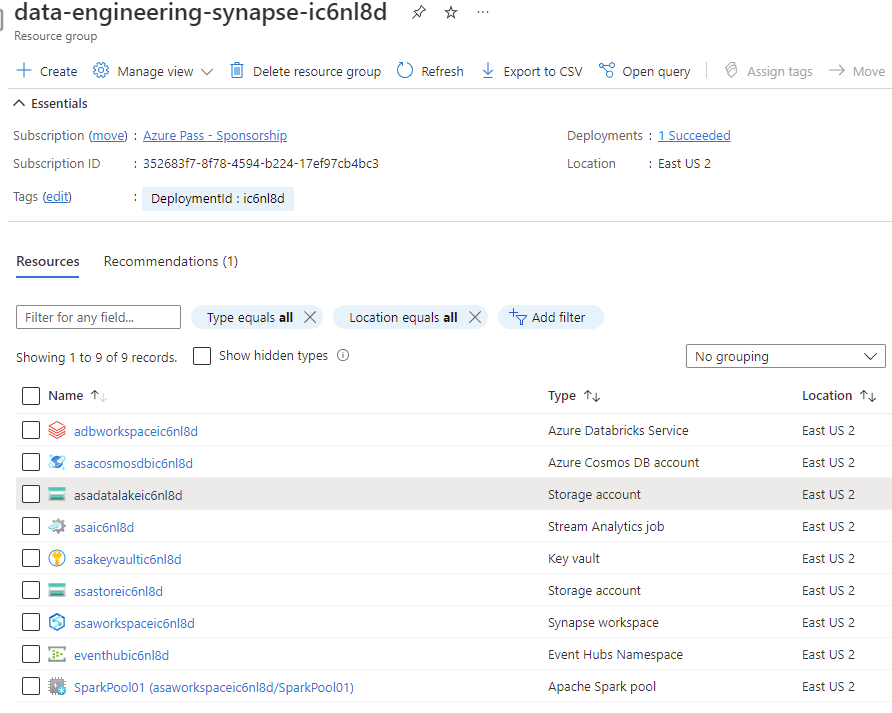
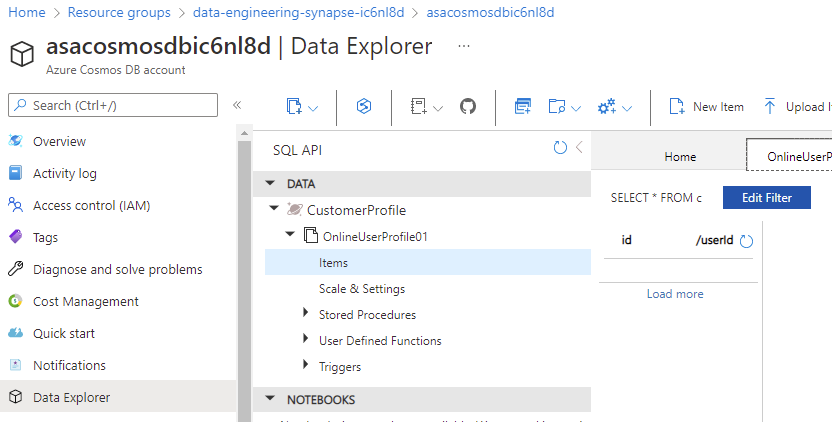
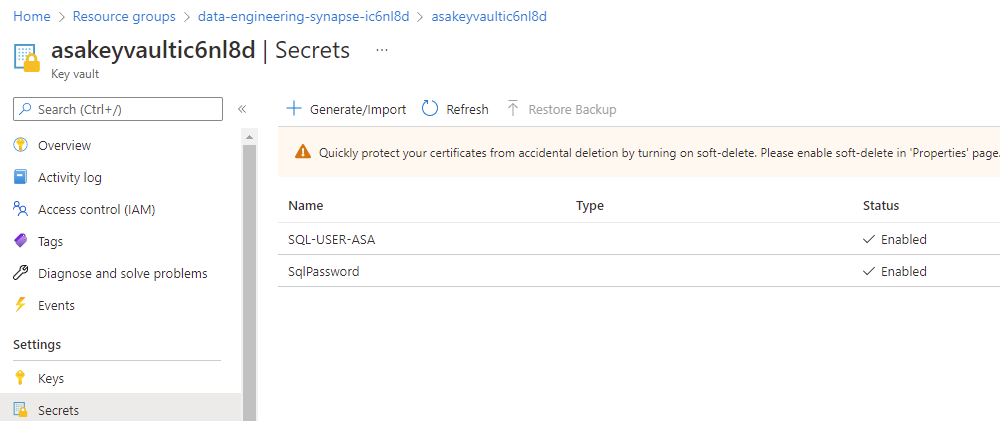
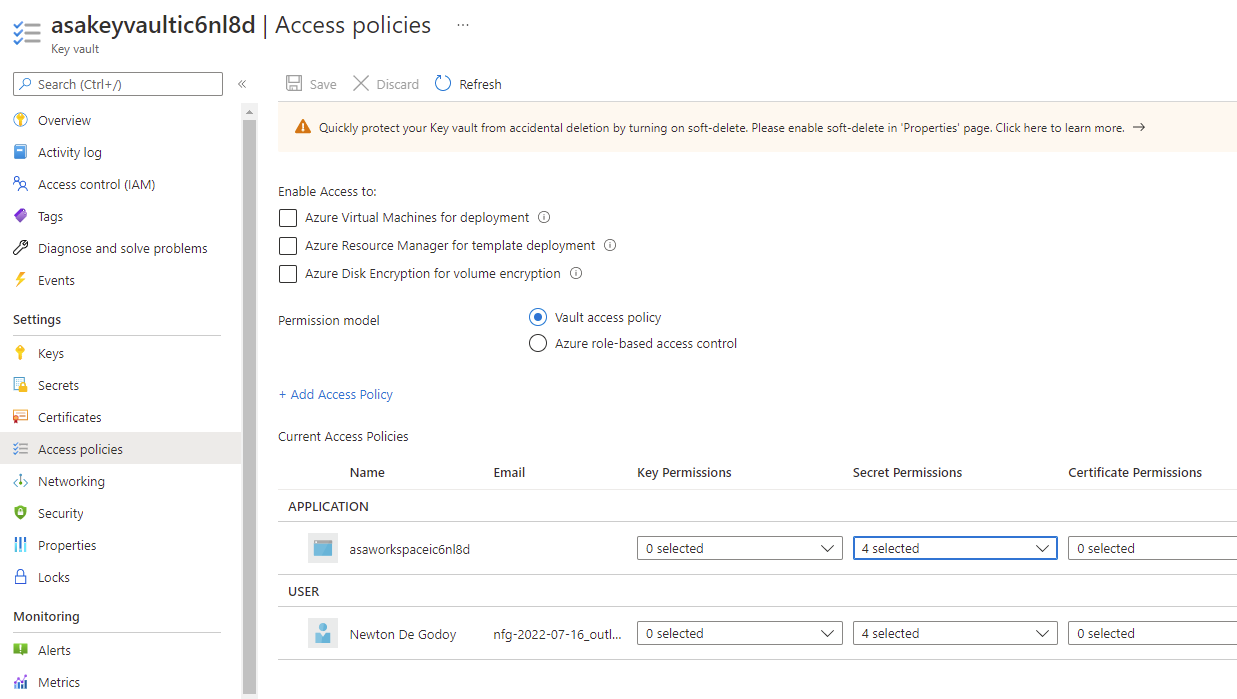
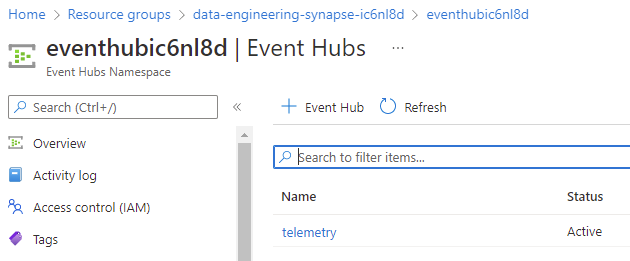
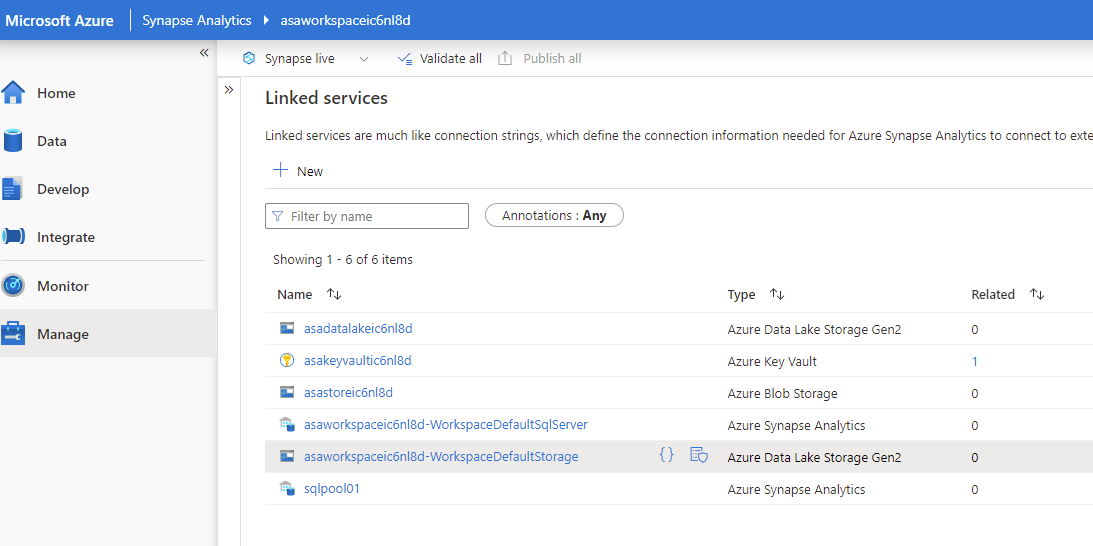
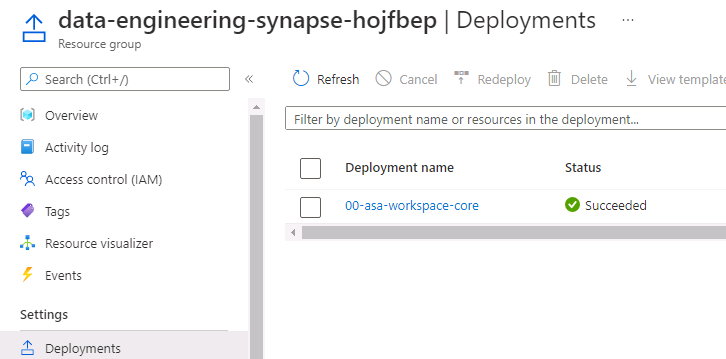


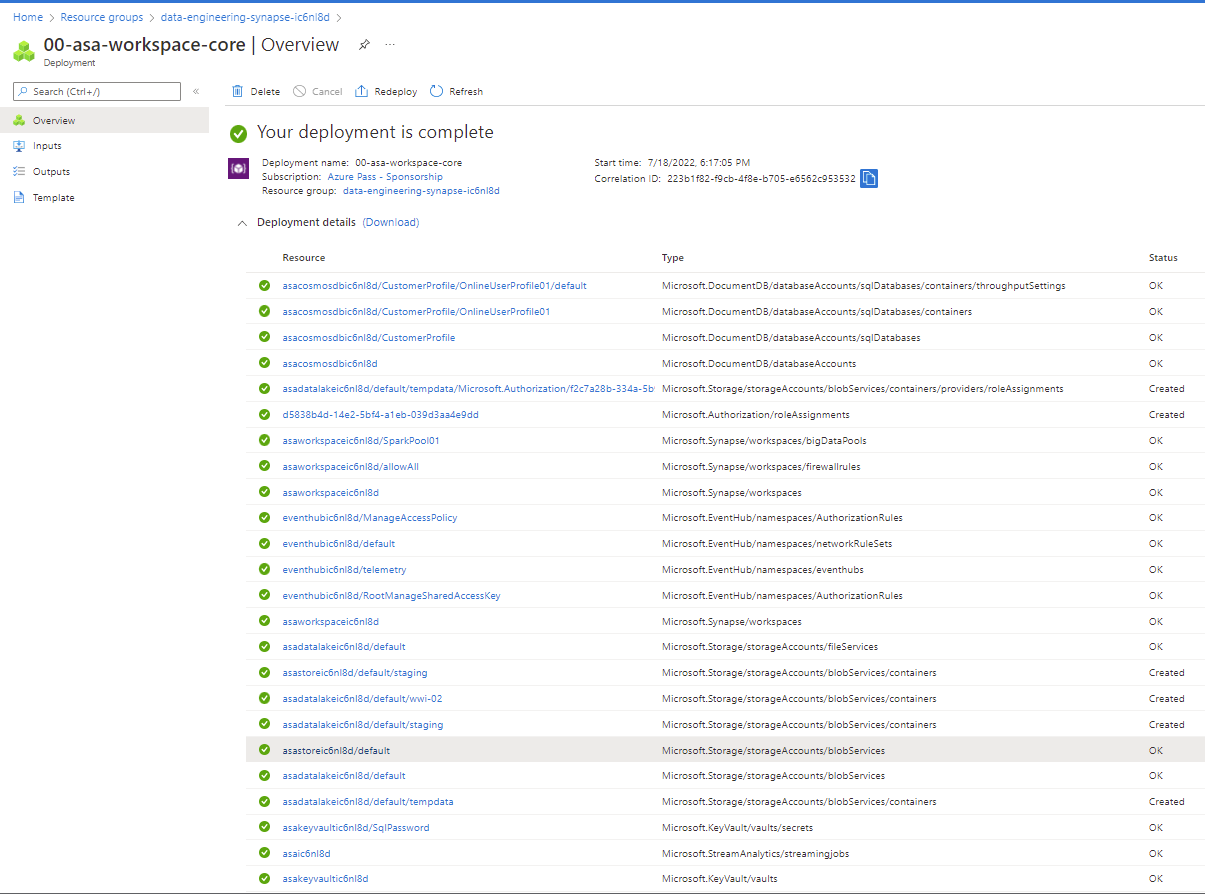
You replace all this by just

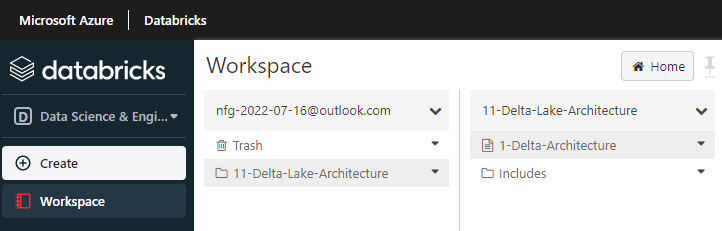
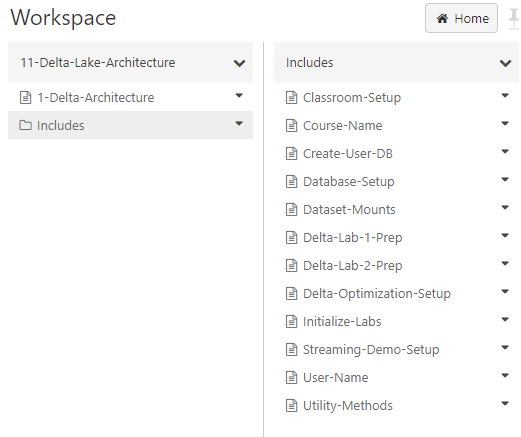
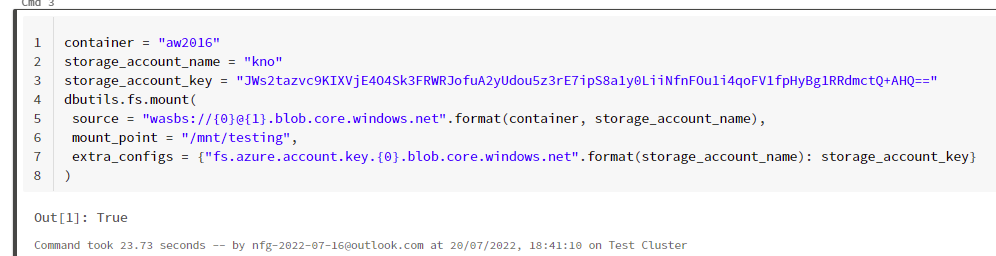
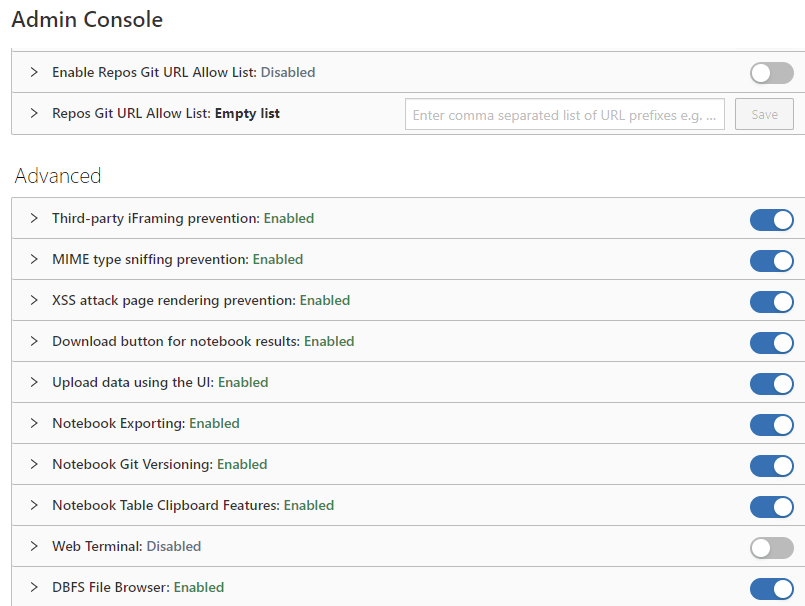
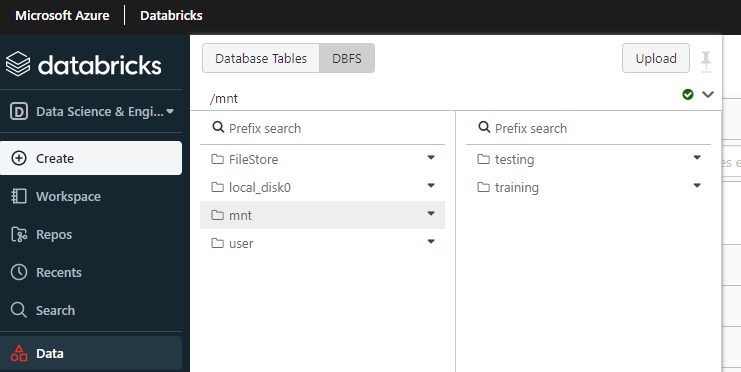
# Add Values from the first setup script here

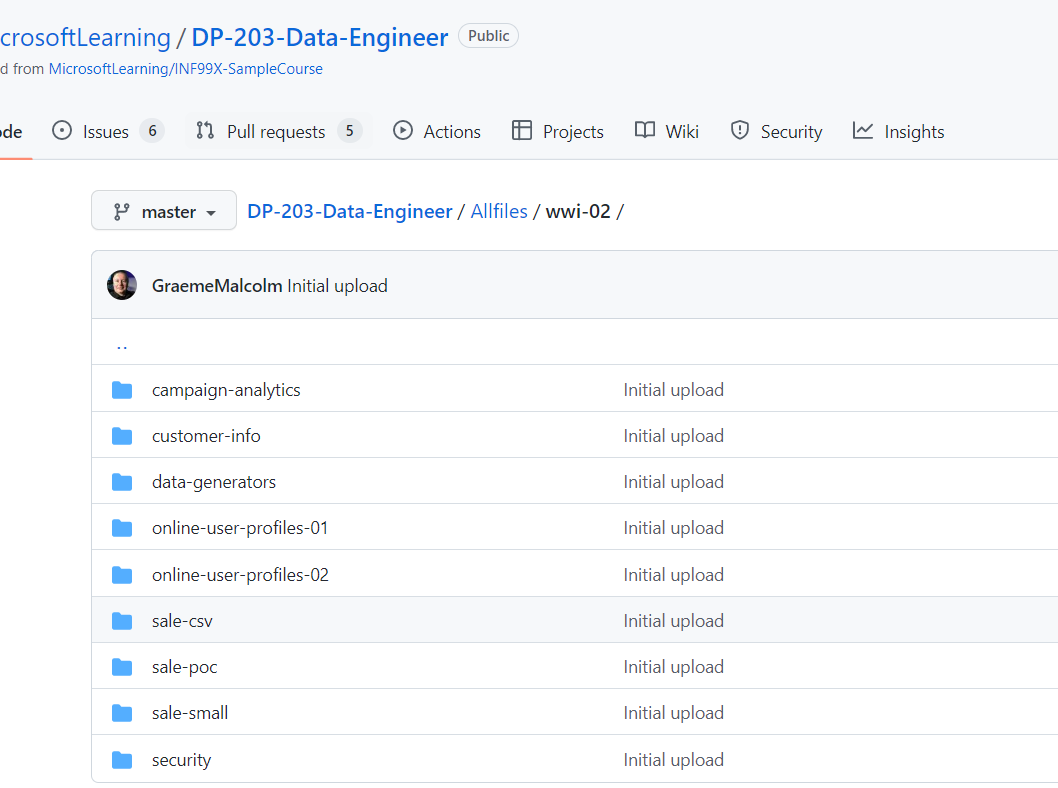
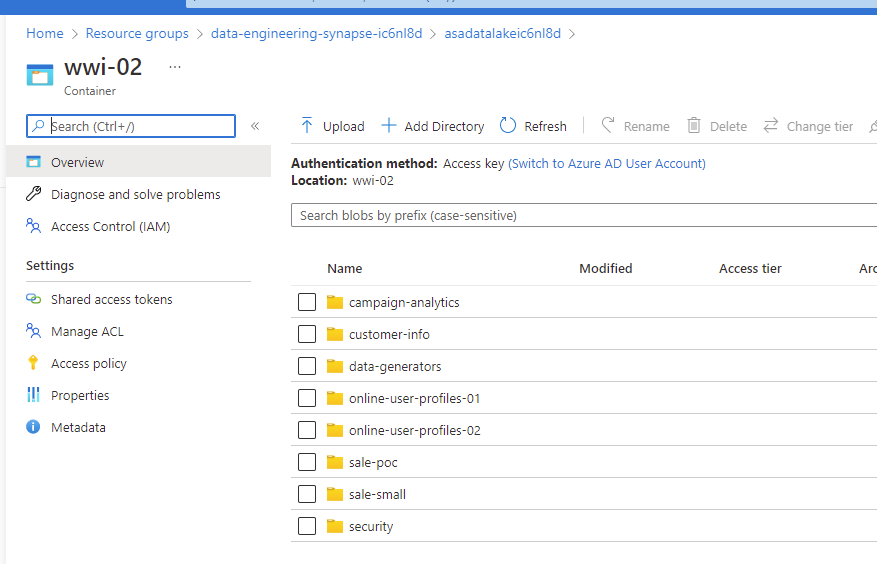
Do the same in the Part03 as well. Only AFTER you make these changes in Part02 and Part03 you will re-execute Part01. You also need to delete the RG in Azure.  
If you execute Part01 a second time BEFORE making the changes to Part02 and Part03, then the workaround is to manually fix the variables in those 2 scripts to have the correct suffix:

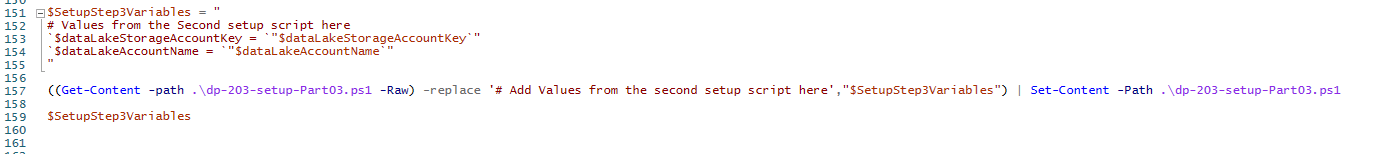
  
Note that is not only line 15. You have the suffix as part of value of other variables in lines 17, 18 and 20.

1. To see what will be created look in c:\dp-203\data-engineering-ilt-deployment\Allfiles\00\artifacts\environment-setup\automation00-asa-workspace-core.json  
     
    (At this stage the container is still empty)    
     
    

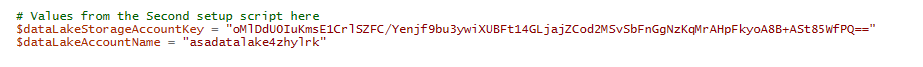


1. I have exported the notebook to d:\dp-203\Lab1\1-Delta-Architecture.ipynb. So it can be opened in Azure Data Studio. Note that we need to import the .dbc file which includes the folder **Includes**.  
      
   In Dataset-Mounts the mount **/mnt/training** is created  
   To create a mount you can use:  
    dbutils.fs.mount(source=”wasbs:// or abfss:// …”, mount\_point = “/mnt/testing”,extra\_configs = { "fs.azure.account.key.{0}.blob.core.windows.net".format(storage\_account\_name): storage\_account\_key}  
     
   You check the existing mounts with **display(dbutils.fs.mounts())**Tounmount, **dbutils.fs.unmount('/mnt/testing')**To browse the DBFS it has to be enabled in Settings | Admin Console | Workspace settings | Advanced | DBFS File Browser  
      
   Read the article <https://docs.microsoft.com/en-us/azure/databricks/data/databricks-file-system> to learn more about DBFS

1. After executing .\dp-203-setup-Part02.ps1, **azcopy** will have copied a lot of files to the datalake. The source is in AllFiles. In wwi-02\online-user-profiles-01 there are 100000 JSON files.  
    
2. Be aware that Part02 will write variables to Part03.



So, if you need to re-execute Part02 you have to replace

  
 By  
 # Add Values from the second setup script here   
before you run Part02

1. I exported the hyperspace notebook. So, it can be imported.