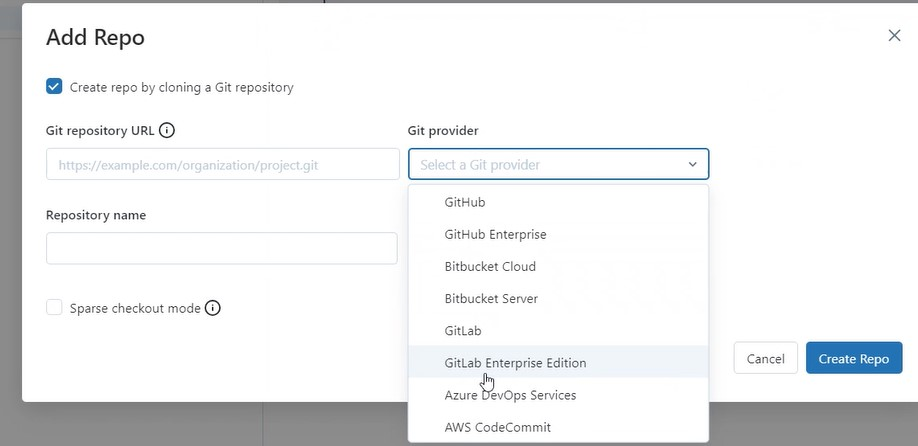
Databricks

Introduction

* It is a managed service
* It provides the backend resources for some service(s)
* It provides a Spark Engine
* To execute something in Spark, a computing unit is provided
* Azure provides the computing unit in this case
* Hence the services we will be using is Azure Databricks

Azure Databricks

* All notebooks to be placed inside repo instead of workflows



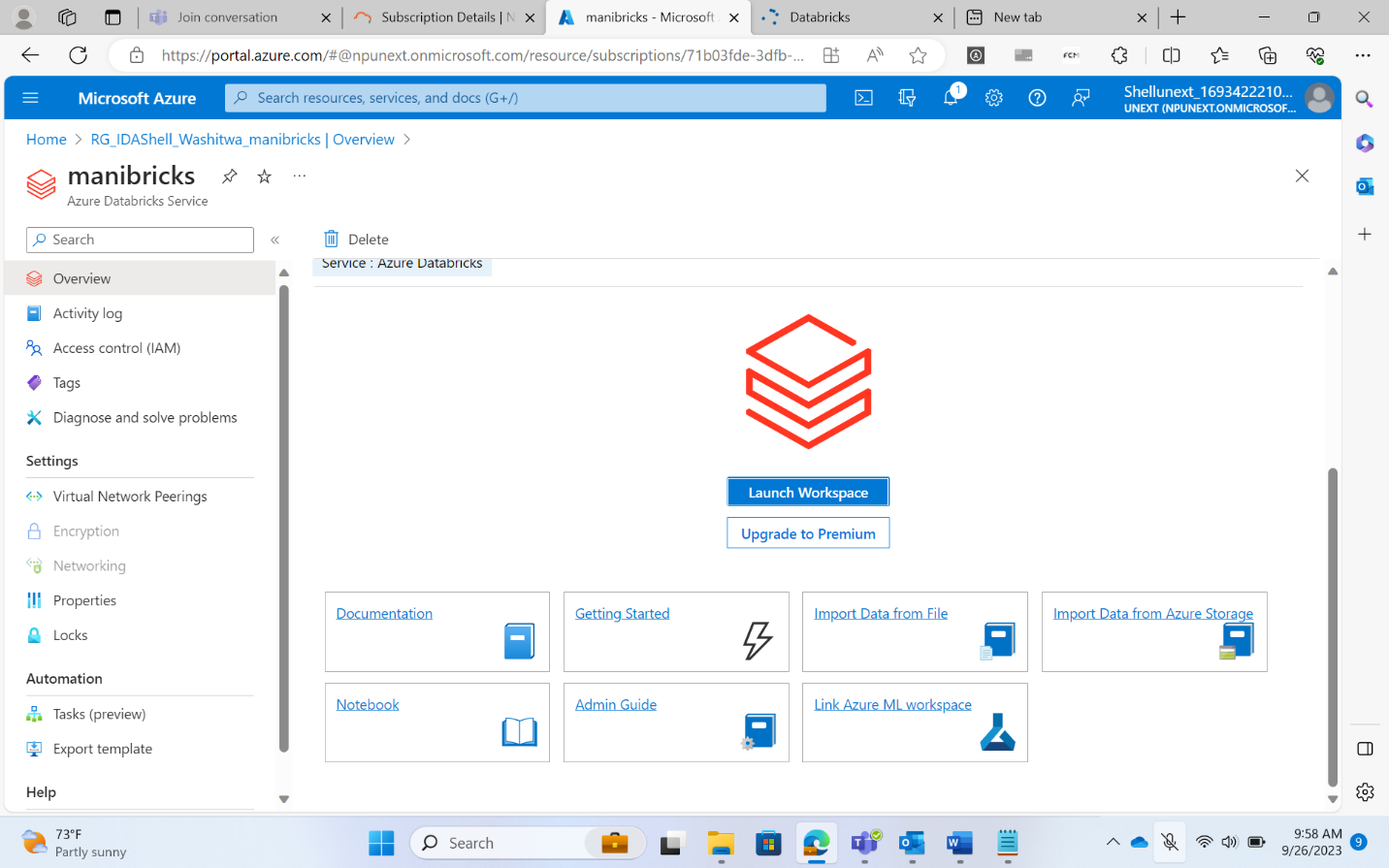
* Job
  + To monitor actions
* Compute
  + To run a cluster, we need compute
  + Multiple clusters can be created and used
* SQL Editor
  + To write SQL codes
* Queries
* Dashboards
* Alerts
  + To create your own alerts
* Query History
* SQL Warehouse
  + To run SQL queries
* Job runs
* Data Ingestion
* Data live tools
  + To work with live data
* Experiments
* Features

Compute

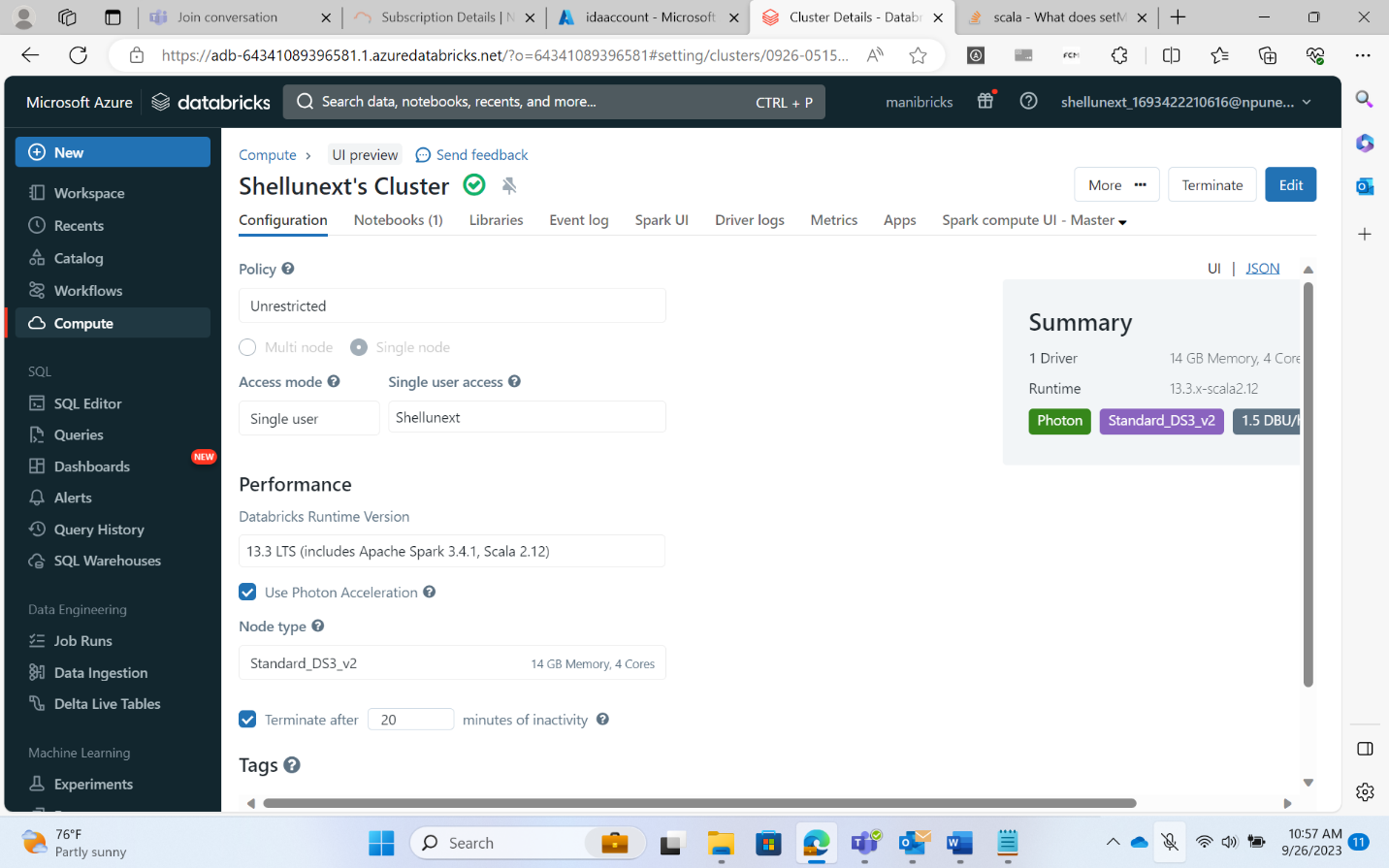
* All-purpose compute
  + Click on create compute
  + Performance: runtime: 13.3 LTS (Long Term Support)
  + Worker type: standard ds3 v2
  + Min worker 2
  + Max worker 8
  + Driver type general purpose
  + The disadvantage of an all-purpose cluster:
    - It must be running all the time
    - If an activity has to be run only for some time periodically, then it is a highly costly approach
* Job compute
  + The cluster starts and terminates automatically
  + It only runs during the scheduled times and not always unlike the all-purpose compute
  + This makes it very cost effective for tasks that need to run at specific times
* SQL Warehouse
* Pools
  + Starting a cluster takes time
  + A cluster pool contains a series of clusters
  + Each cluster has a number of VM’s allocated
  + If a cluster is created in all-purpose compute, the VM’s are allocated from here
  + This reduces the cluster start time and makes the computation faster

To Do

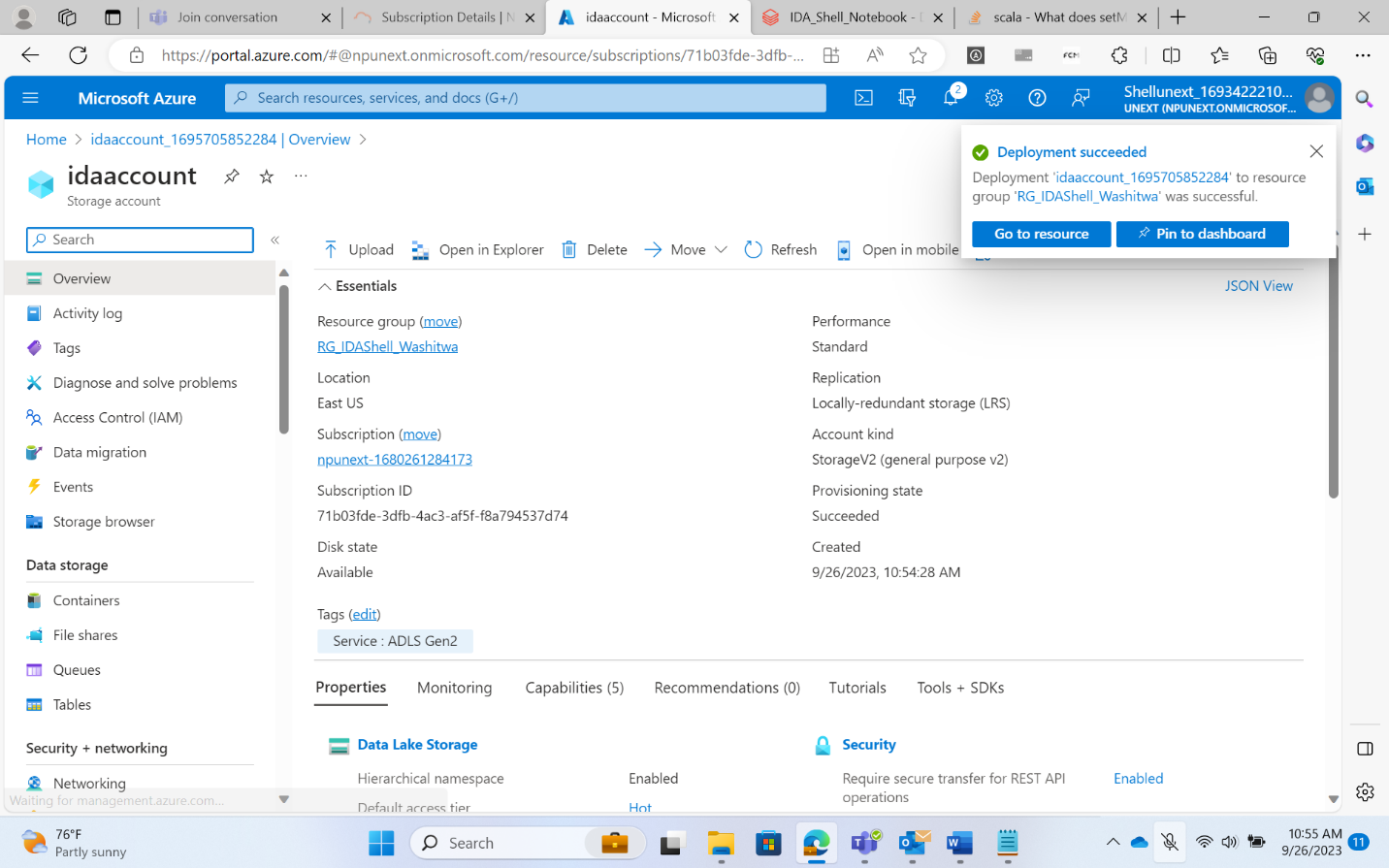
* Mount Storage Account to Databricks
* Instead of reading data using paths and links which are complicated, the entire Storage Account can be mounted in Databricks to make the data imports easier
* Create a Azure Databricks account



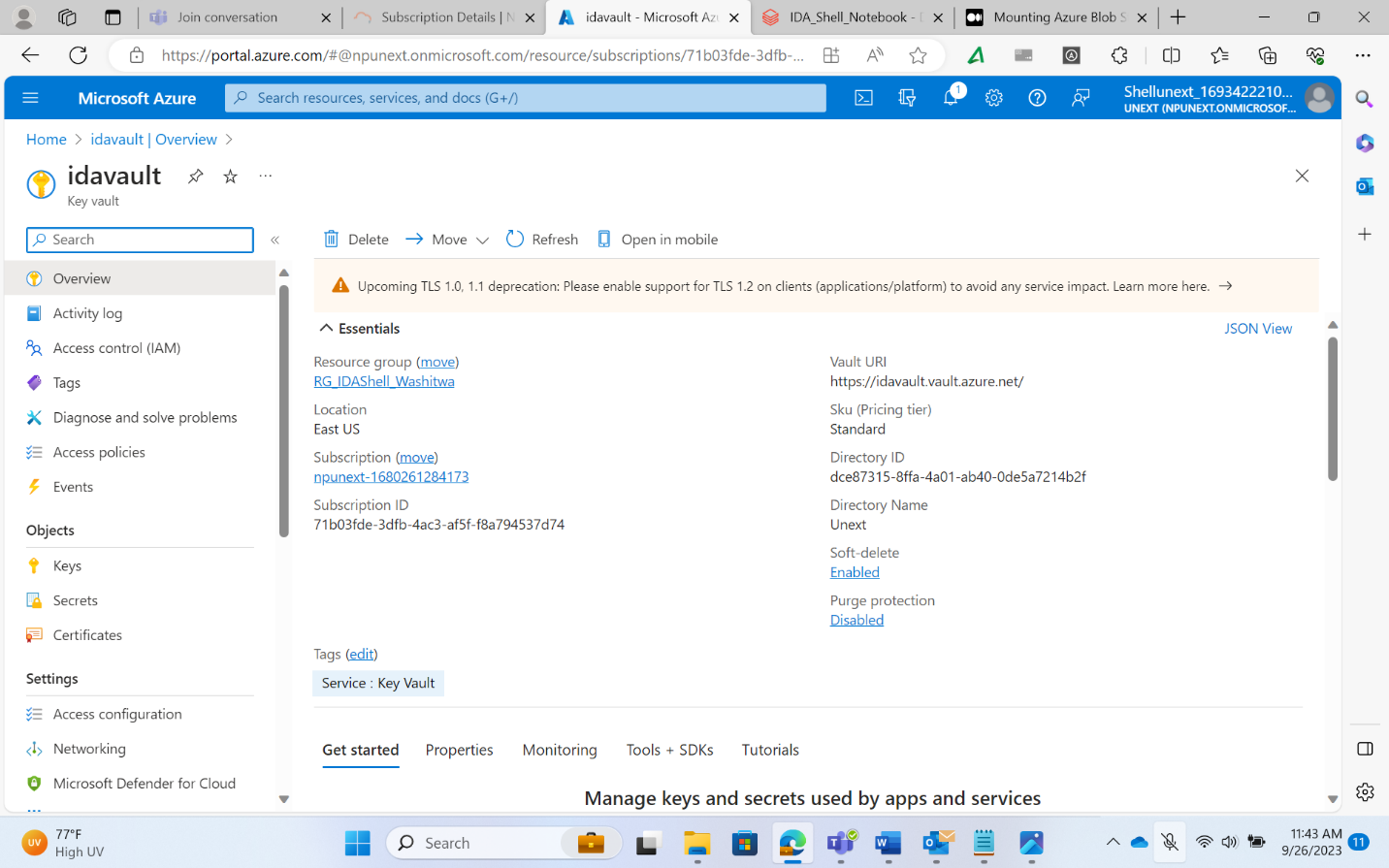
* Create and start an All-Purpose Compute Cluster



* Create a storage account

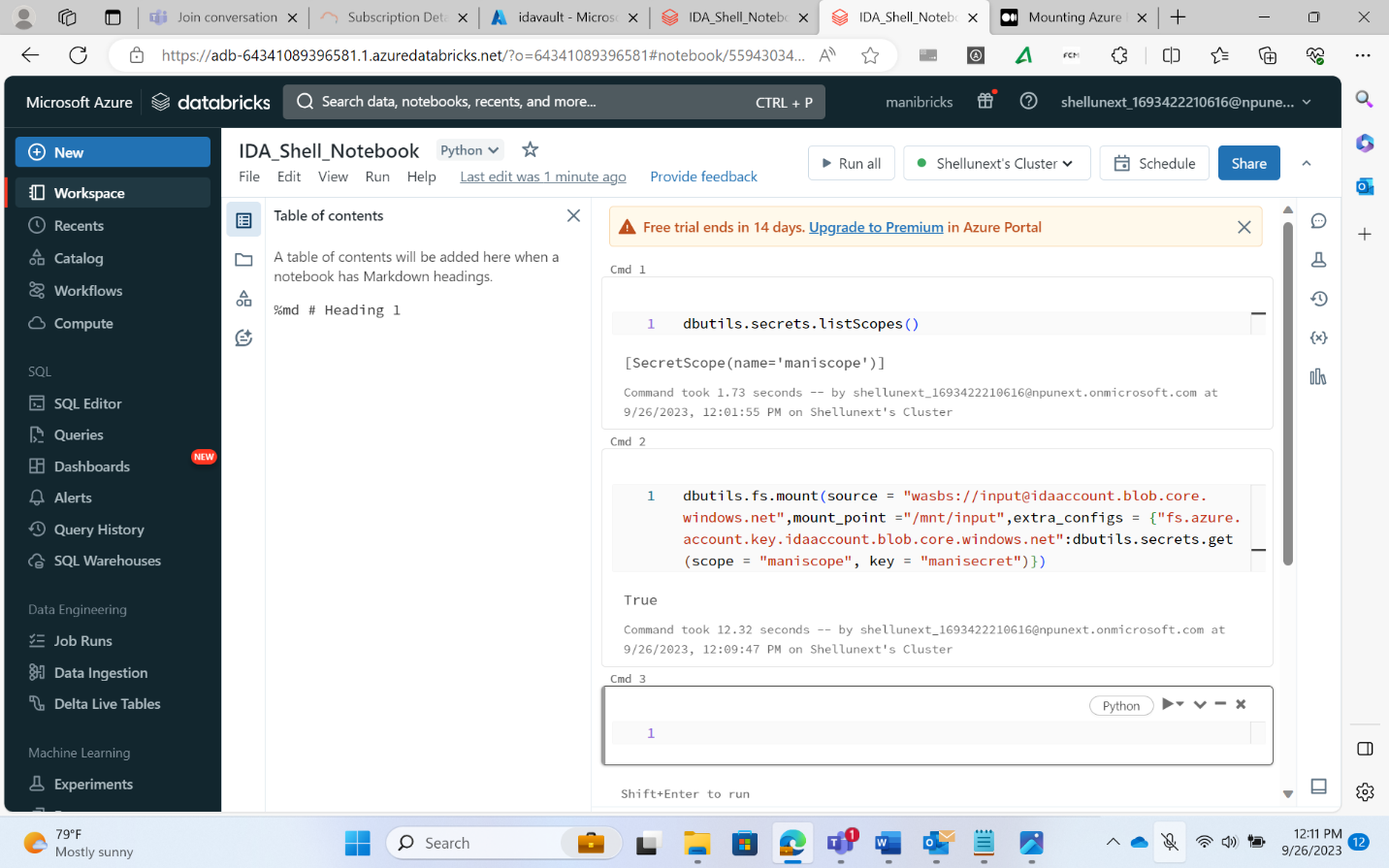


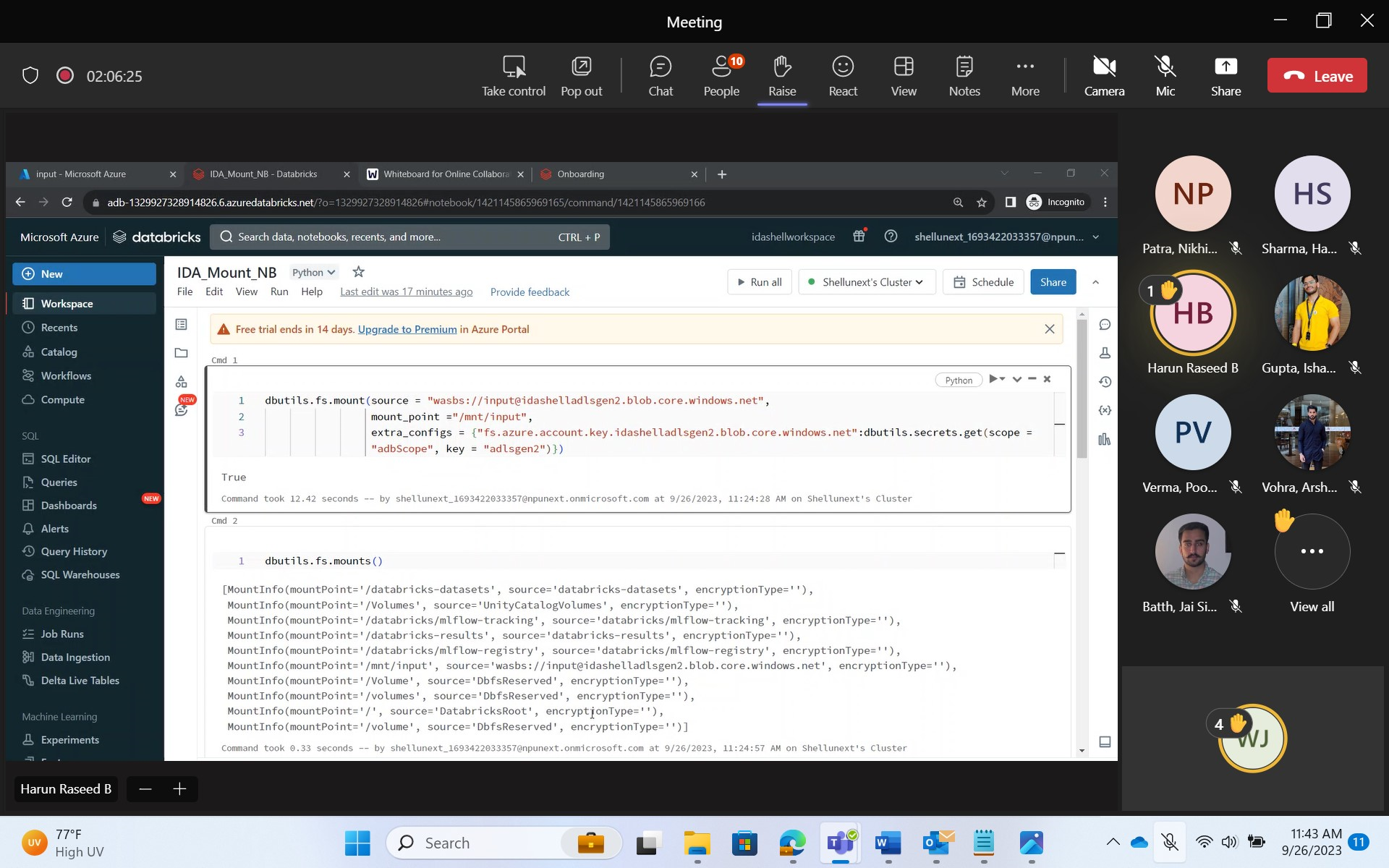
Create Key Vault



dbutils.fs.mount(source = "wasbs://input@idaaccount.blob.core.windows.net",mount\_point ="/mnt/input",extra\_configs = {"fs.azure.account.key.idaaccount.blob.core.windows.net":dbutils.secrets.get(scope = "maniscope", key = "manisecret")})

Mount the Storage Account





Widgets

* For parameterization
* dbutils.widgets.text(“test”, “ida”)
* dbutils.widgets.get(“test”)
* dbutils.widgets.removeAll()
* dbutils.widgets.dropdown(“choose\_colors”, “white”, [“Red”, “black”, “blue”], “color\_dropdown”)
* instead of df.show(), use df.display() in Databricks
* the default table provider in Databricks is Delta

SQL Queries

Utility Notebook

Writing Data as a Table(Managed and External)

Streaming

* process used for writing is called write stream
* process used for reading is called read stream

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Structured Streaming

\*\*

from pyspark.sql.types import \*

schema = StructType([StructField("lsoa\_code", StringType(), True),\

StructField("borough", StringType(), True),\

StructField("major\_category", StringType(), True),\

StructField("minor\_category", StringType(), True),\

StructField("value", StringType(), True),\

StructField("year", StringType(), True),\

StructField("month", StringType(), True)])

Streamdf = spark.readStream.schema(schema).option("header",True).csv("/mnt/input/droplocation")

trimmedDF = Streamdf.select(

Streamdf.borough,

Streamdf.year,

Streamdf.month,

Streamdf.value

)\

.withColumnRenamed(

"value",

"convictions"

)

query = trimmedDF.writeStream\

.outputMode("append")\

.format("csv") \

.option("path", "/mnt/input/destination") \

.option("checkpointLocation", "/mnt/input/checkpoint") \

.start()\

.awaitTermination()