**ASSIGNMENT-10**

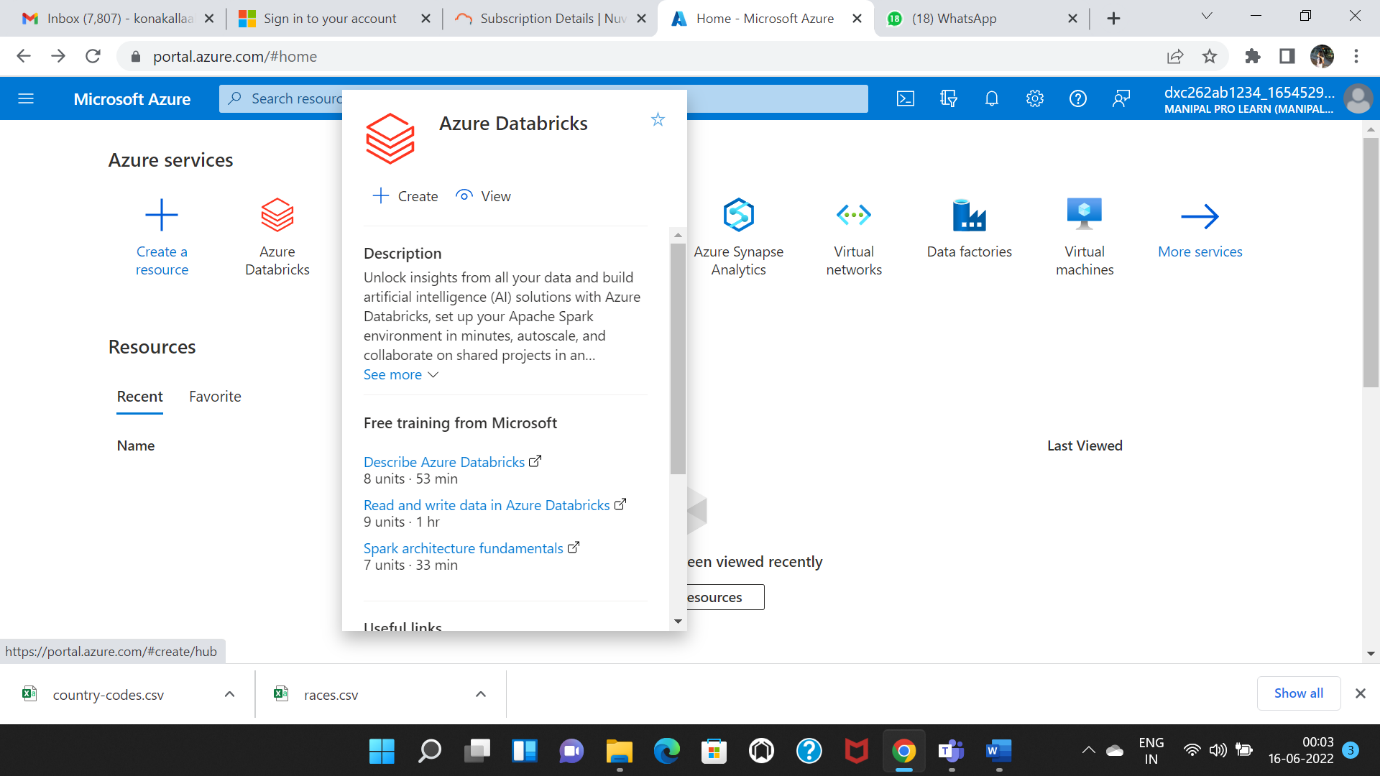
NAME – KONAKALLA Akhila

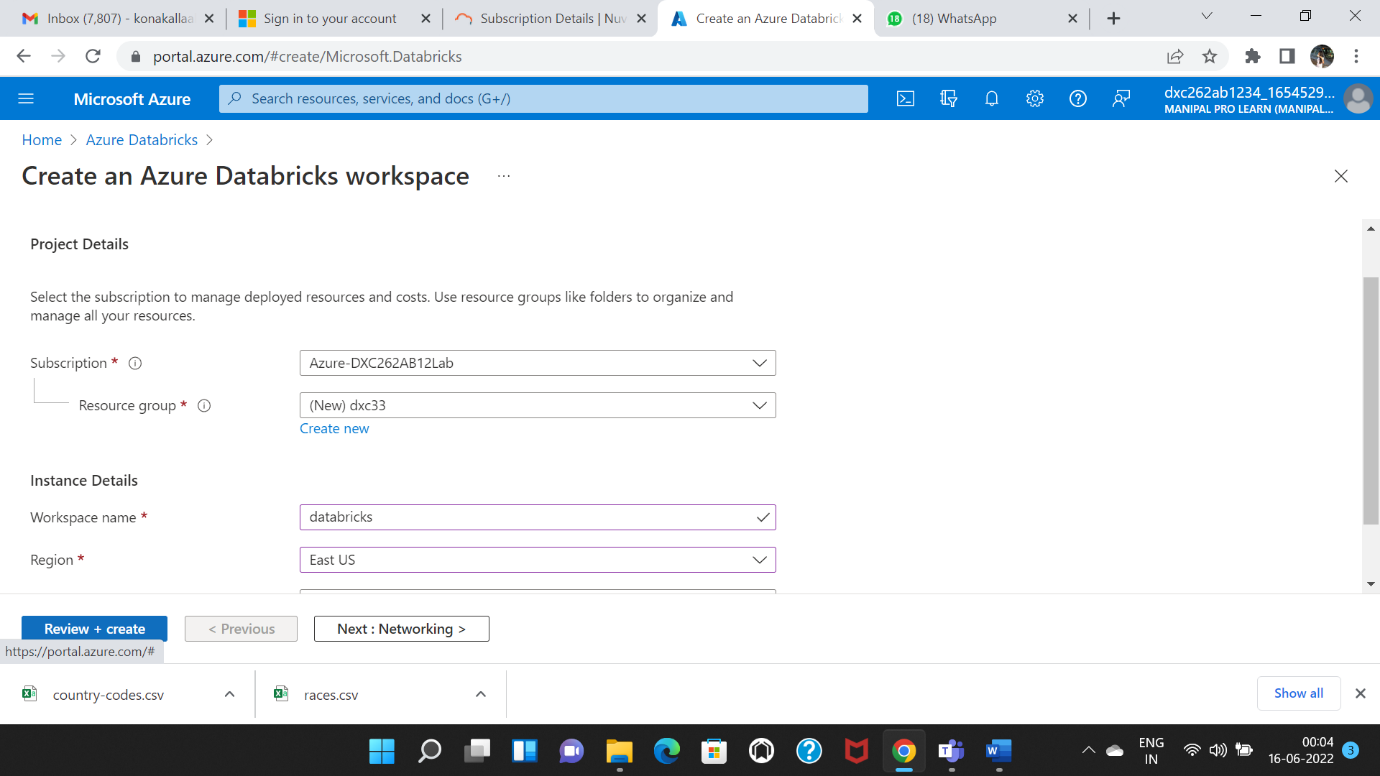
BATCH – DXC-262-ANALYTICS-B12-AZURE

DATE OF SUBMISSION – 15th JUNE 2022

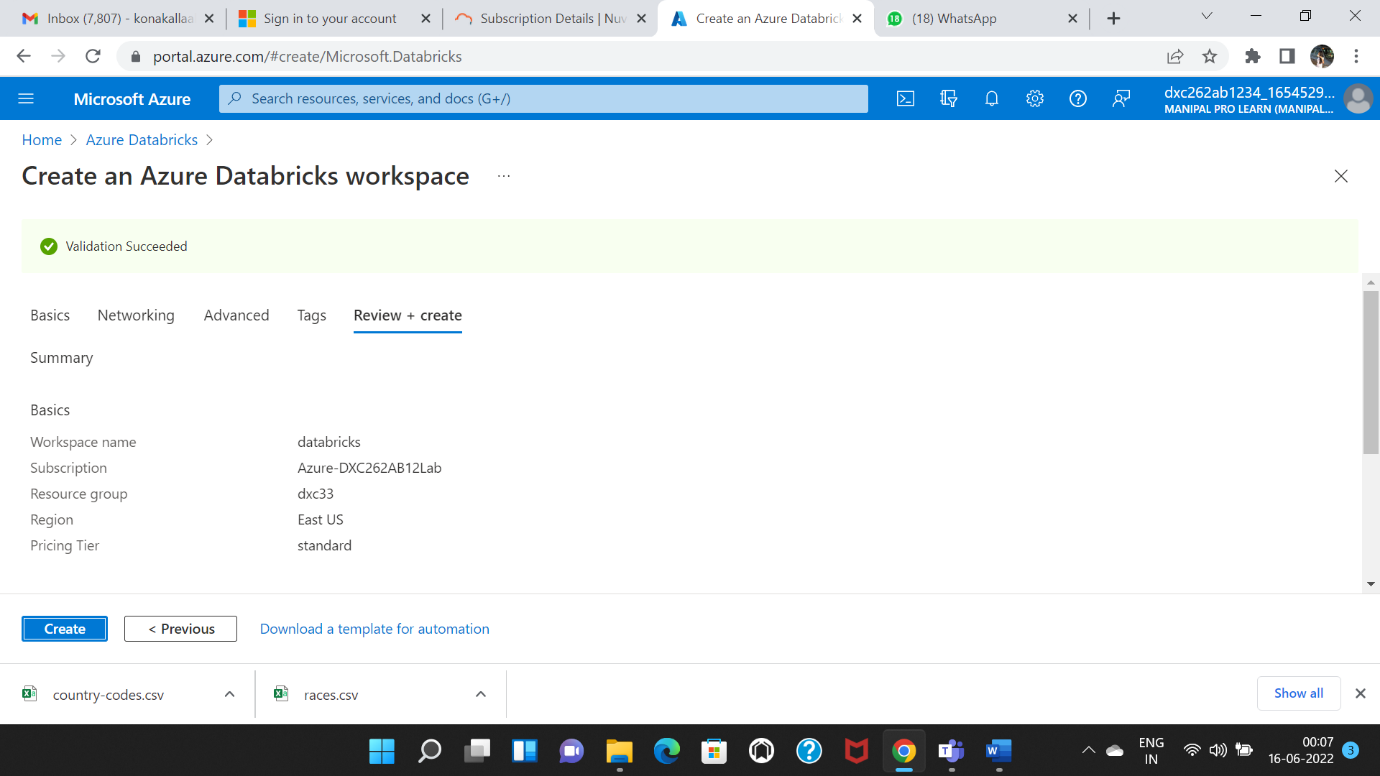
1.Using archive1.zip file – Please ingest data into databricks DBFS path & query the data, redesign columns, accordingly using dataframe commands – display with notebooks accordingly ?

1.First create a databricks.

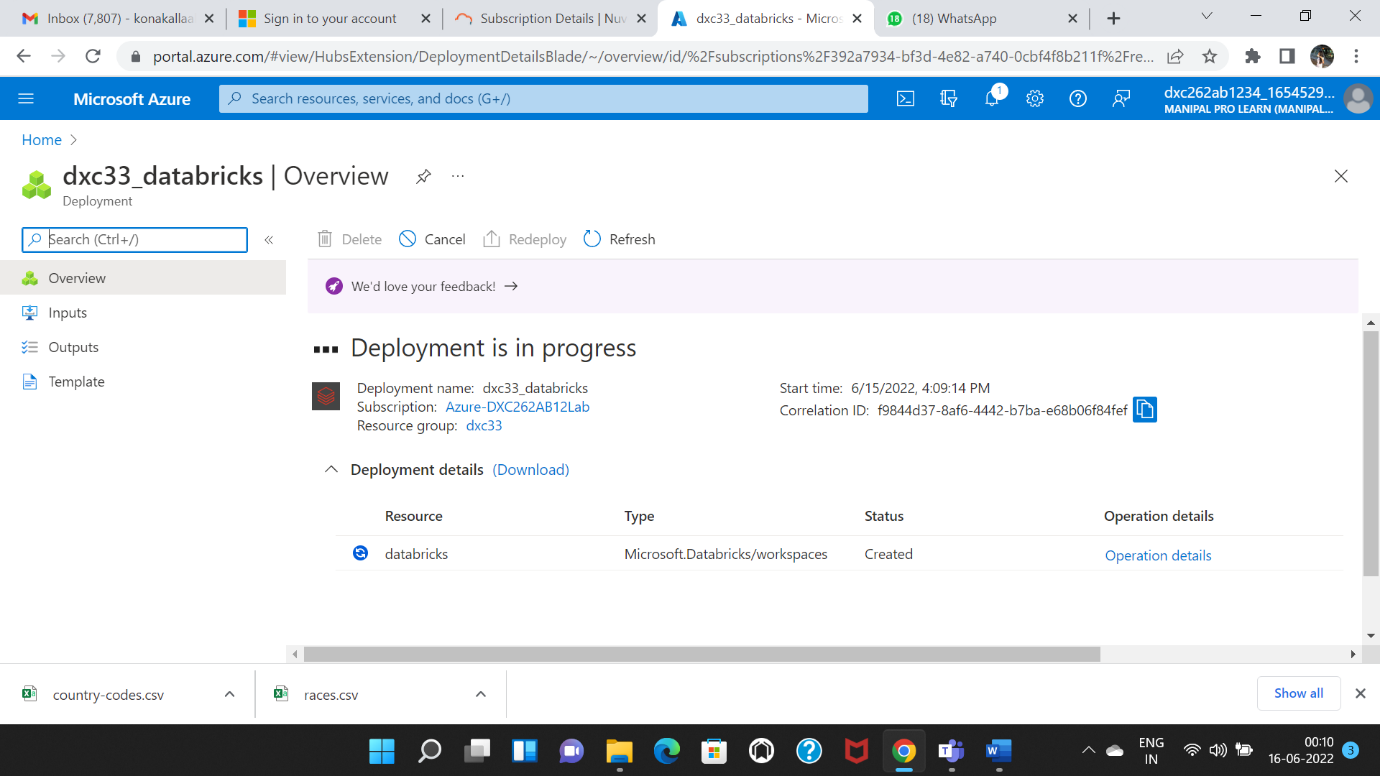


2.Click on Review+create

3.Click on create.

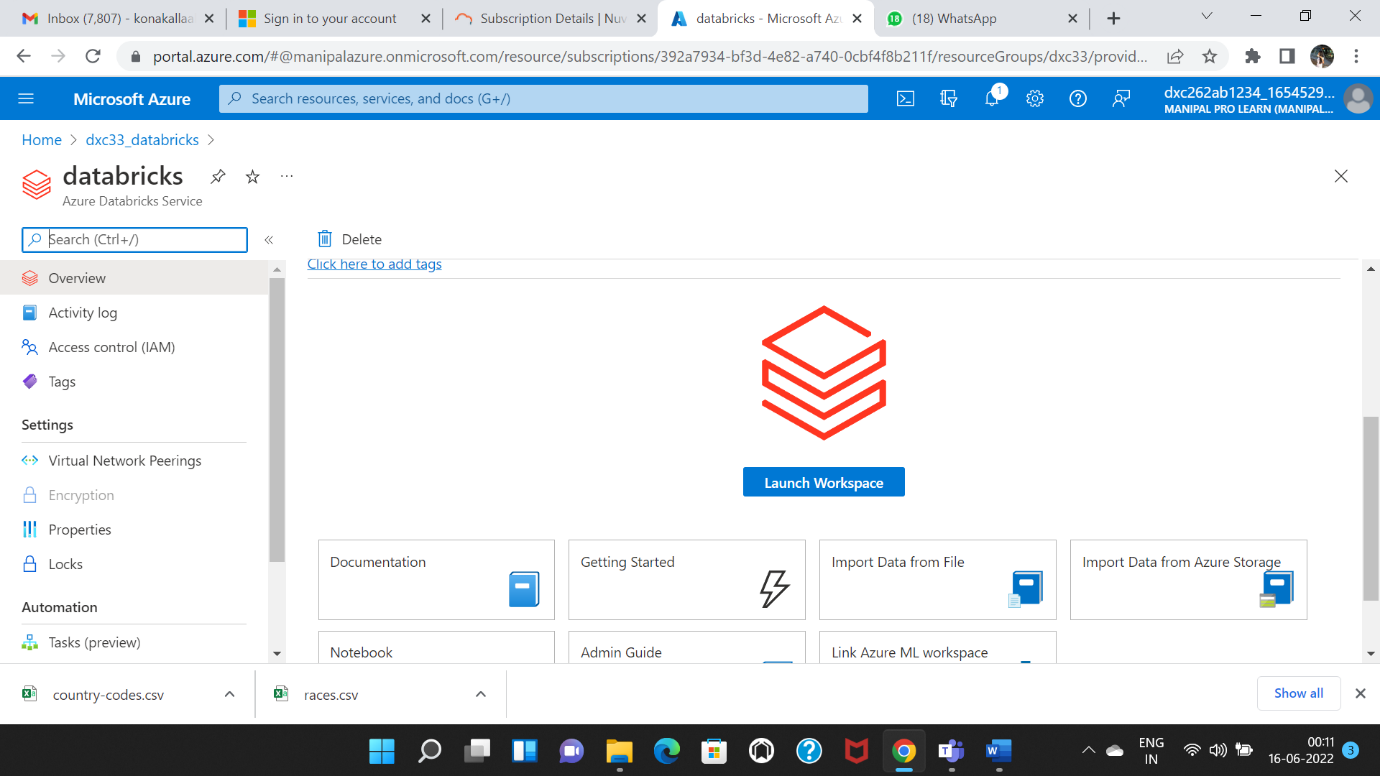


4.Deploying the process starts.

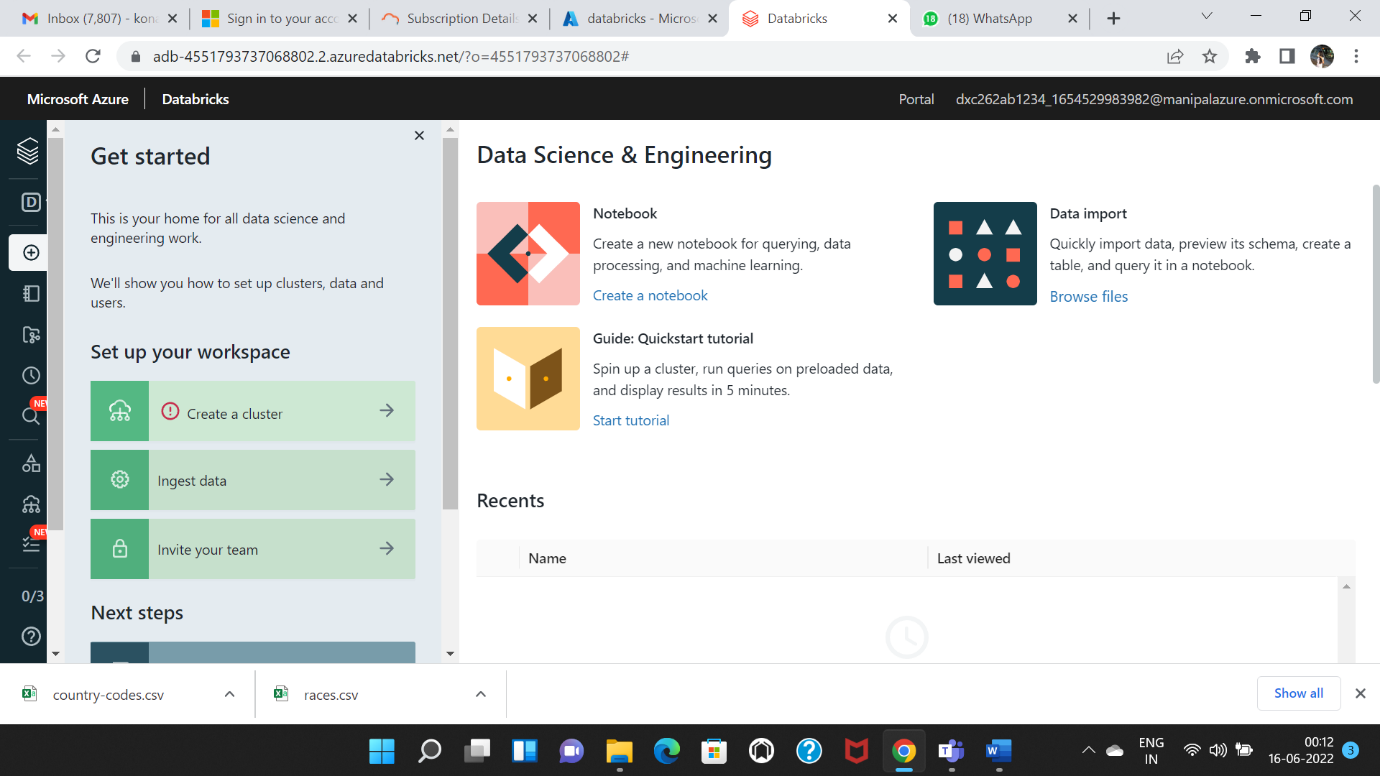


5. Click on Go to resources

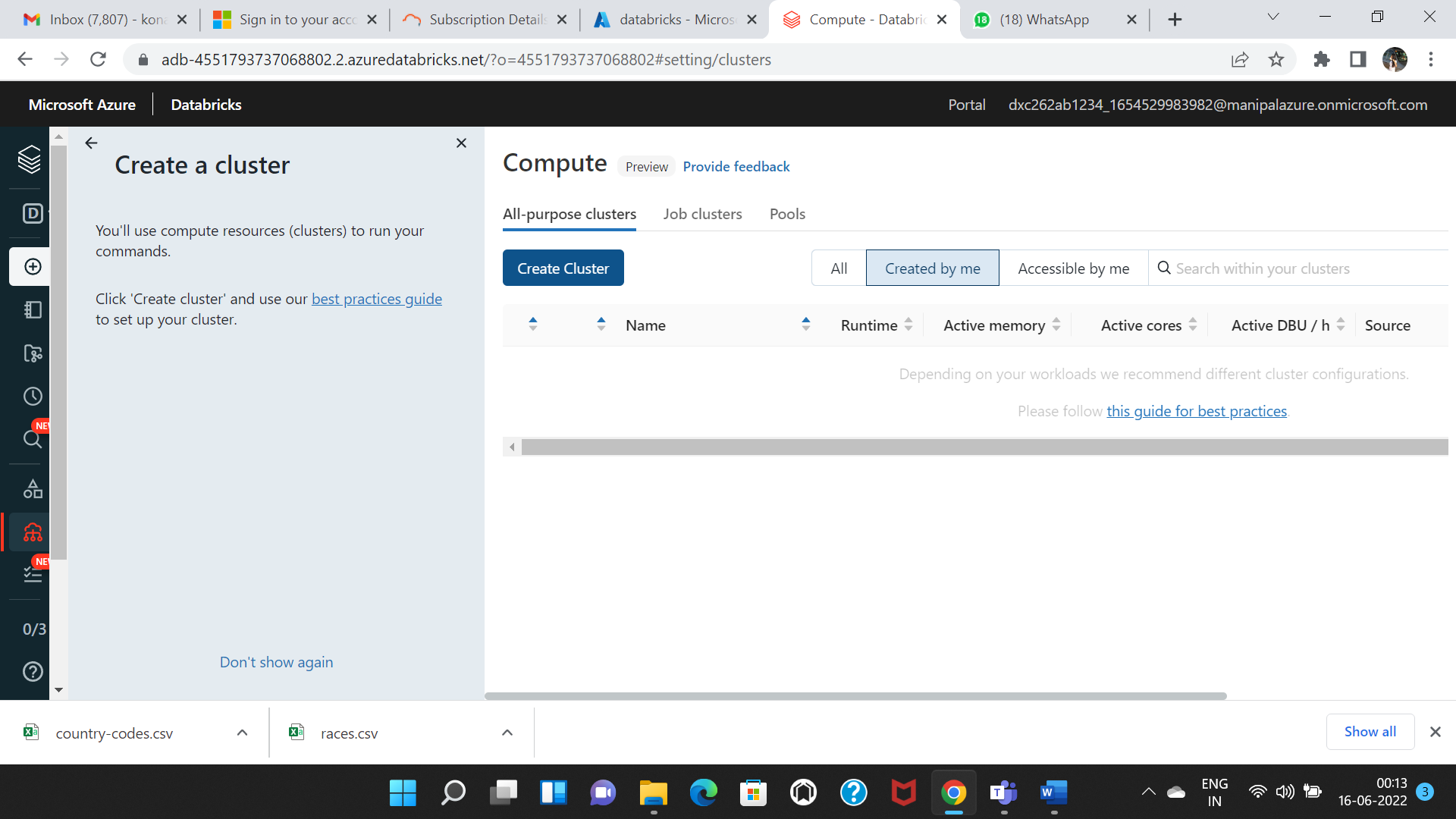
6.Click on workspace.



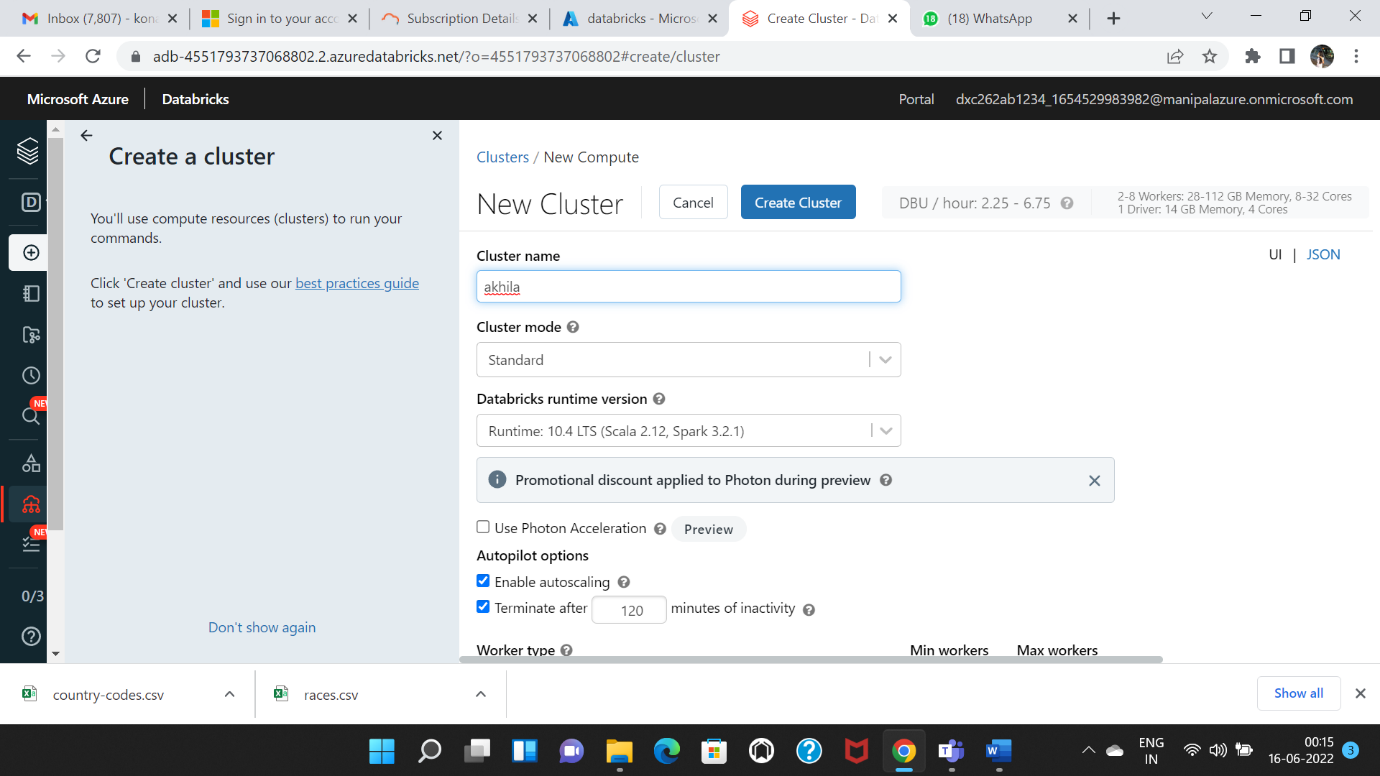
7.Click on create cluster.



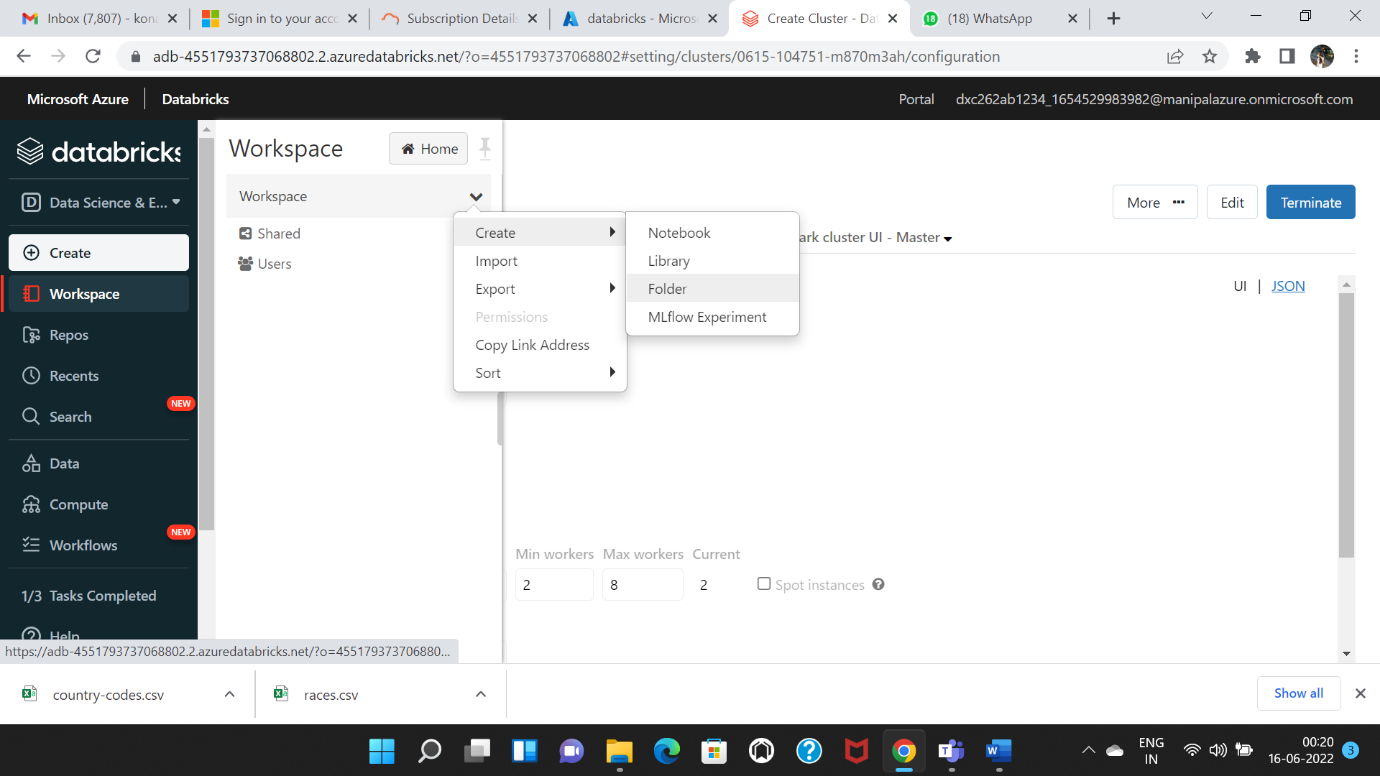
8.Click on create cluster.



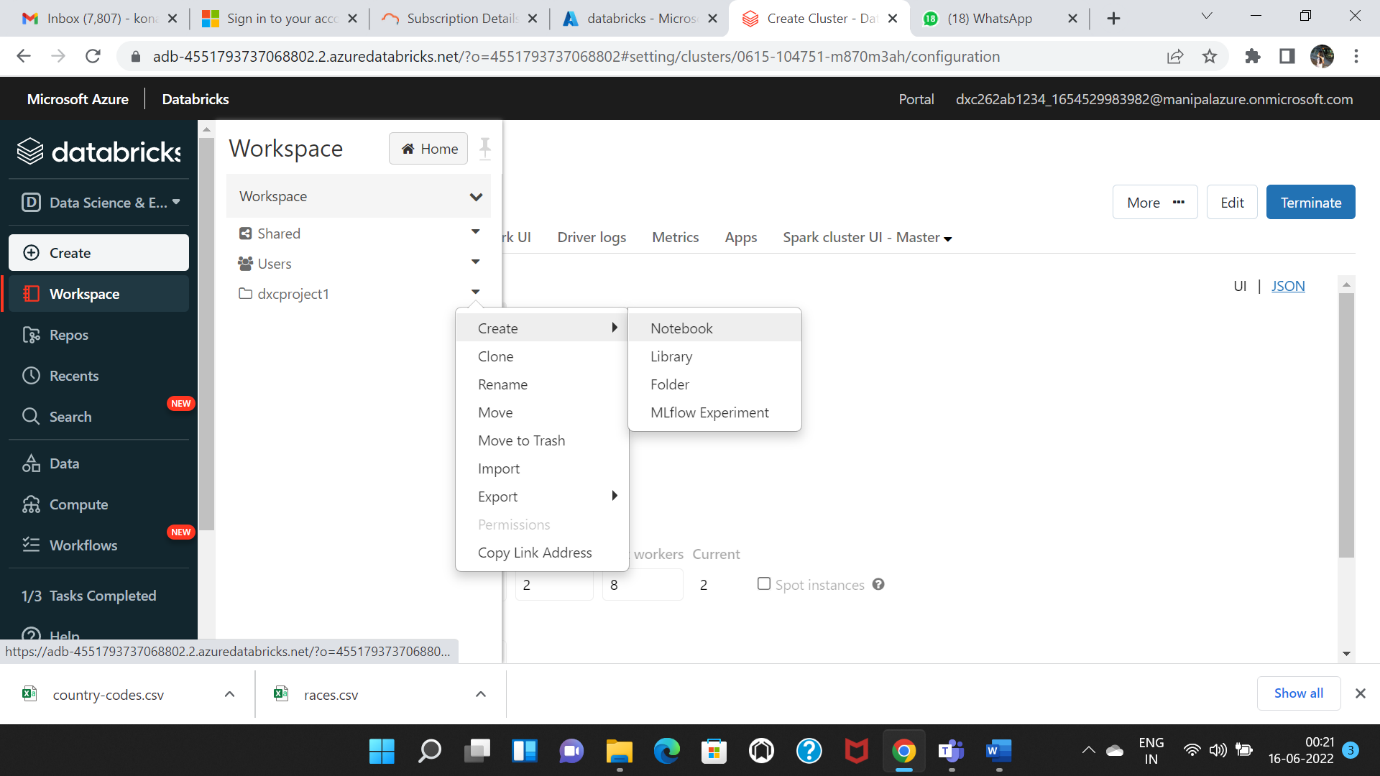
9.Create a cluster name



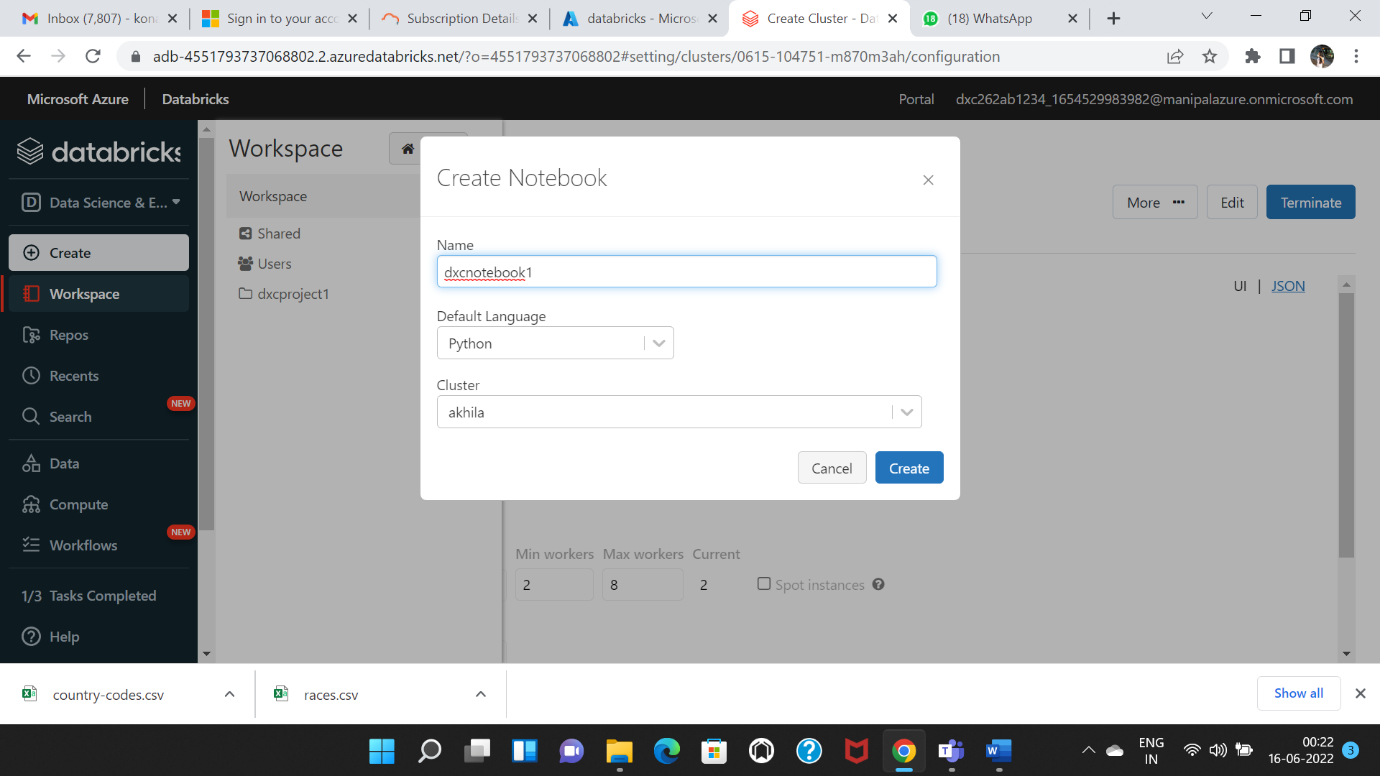
10.Click on workspace create a folder.



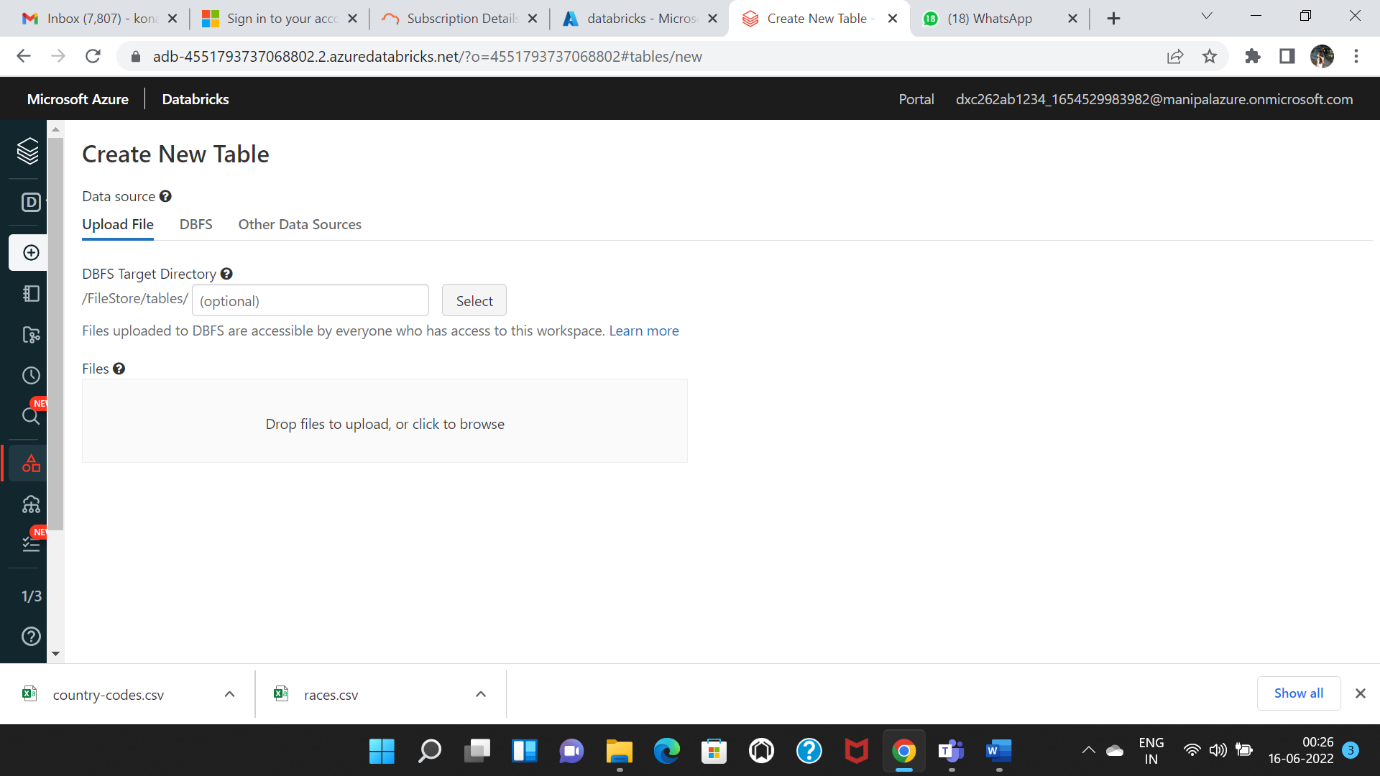
11.After creating folder again create a notebook for that folder.



12. Create notebook



13.Upload the cvs file by clicking on the data and click on create table.



14. Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType

Graphical user interface, text, application, email

Description automatically generated

country\_codes\_schema = StructType(fields=[StructField("FIFA", StringType(),False),

StructField("Dial", StringType(),True),

StructField("ISO3166-1-Alpha-3",StringType(),True),

StructField("MARC", StringType(),True),

StructField("is\_independent", StringType(),True),

StructField("ISO3166-1-numeric",IntegerType(),True),

StructField("GAUL", IntegerType(),True),

StructField("FIPS", StringType(),True),

StructField("WMO", StringType(),True),

StructField("ISO3166-1-Alpha-2",StringType(),True),

StructField("ITU", StringType(),True),

StructField("IOC", StringType(),True),

StructField("DS", StringType(),True),

StructField("UNTERM Spanish Formal", StringType(),True),

StructField("Global Code",StringType(),True),

StructField("Intermediate Region Code",IntegerType(),True),

StructField("official\_name\_fr",StringType(),True),

StructField("UNTERM French Short",StringType(),True),

StructField("ISO4217-currency\_name",StringType(),True),

StructField("Developed / DevelopingCountries", StringType(),True),

StructField("UNTERM Russian Formal",StringType(),True),

StructField("UNTERM English Short",StringType(),True),

StructField("ISO4217-currency\_alphabetic\_code",StringType(),True),

StructField("Small Island Developing States (SIDS)",StringType(),True),

StructField("UNTERM Spanish Short",StringType(),True),

StructField("ISO4217-currency\_numeric\_code",IntegerType(),True),

StructField("UNTERM Chinese Formal",StringType(),True),

StructField("UNTERM French Formal",StringType(),True),

StructField("UNTERM Russian Short",StringType(),True),

StructField("M49",IntegerType(),True),

StructField("Sub-region Code",IntegerType(),True),

StructField("Region Code",IntegerType(),True),

StructField("official\_name\_ar",StringType(),True),

StructField("ISO4217-currency\_minor\_unit",IntegerType(),True),

StructField("UNTERM Arabic Formal",StringType(),True),

StructField("UNTERM Chinese Short",StringType(),True),

StructField("Land Locked Developing Countries (LLDC)",StringType(),True),

StructField("Intermediate Region Name",StringType(),True),

StructField("official\_name\_es",StringType(),True),

StructField("UNTERM English Formal",StringType(),True),

StructField("official\_name\_cn",StringType(),True),

StructField("official\_name\_en",StringType(),True),

StructField("ISO4217-currency\_country\_name",StringType(),True),

StructField("Least Developed Countries (LDC)",StringType(),True),

StructField("Region Name",StringType(),True),

StructField("UNTERM Arabic Short",StringType(),True),

StructField("Sub-region Name",StringType(),True),

StructField("official\_name\_ru",StringType(),True),

StructField("Global Name",StringType(),True),

StructField("Capital",StringType(),True),

StructField("Continent",StringType(),True),

StructField("TLD",StringType(),True),

StructField("Languages",StringType(),True),

StructField("Geoname ID",IntegerType(),True),

StructField("CLDR display name",StringType(),True),

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

A screenshot of a computer

Description automatically generated

country\_codes\_df = spark.read \

.option("header" , True) \

.schema(country\_codes\_schema) \

.csv("/FileStore/tables/country\_codes.csv")

A screenshot of a computer

Description automatically generated

from pyspark.sql.functions import col,lit

country\_codes\_selected\_df = country\_codes\_df.select(col('FIFA'),

col('Dial'),col('Developed / Developing Countries').alias('D/UD'),col('UNTERM Chinese Short').alias('Unterm\_Chinese\_Short'),col('Land Locked Developing Countries (LLDC)').alias('LLDC'),col('official\_name\_es'),col('Region Name'),col('EDGAR'))

Graphical user interface, text, application, email

Description automatically generated

display(country\_codes\_selected\_df)

A screenshot of a computer

Description automatically generated

2. Using archive2.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

First, login to your Azure Portal and create a Databricks workspace.

Open the Databricks workspace and create cluster for your future use.

Now, create a notebook by clicking on the create Notebook option from the side panel.

A screenshot of a computer

Description automatically generated

After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the drag & drop region.

Graphical user interface, application, Word

Description automatically generated

from pyspark.sql.types import StructType, StructField, IntegerType, StringType

Graphical user interface, text, application, email

Description automatically generated

nces330\_20\_schema = StructType(fields=[StructField("year",IntegerType(),False),

StructField("State",StringType(),True),

StructField("Type",StringType(),True),

StructField("Length",StringType(),True),

StructField("Expense",StringType(),True),

StructField("Value",IntegerType(),True),

])

A screenshot of a computer

Description automatically generated

nces330\_20\_df = spark.read \

.option("header" , True) \

.schema(nces330\_20\_schema) \

.csv("/FileStore/tables/nces330\_20.csv")

A screenshot of a computer

Description automatically generated

from pyspark.sql.functions import col,lit

nces330\_20\_selected\_df = nces330\_20\_df.select(col('Year'),col('State'),col('Expense'))

Graphical user interface, text, application, email

Description automatically generated

display(nces330\_20\_selected\_df)

A screenshot of a computer

Description automatically generated

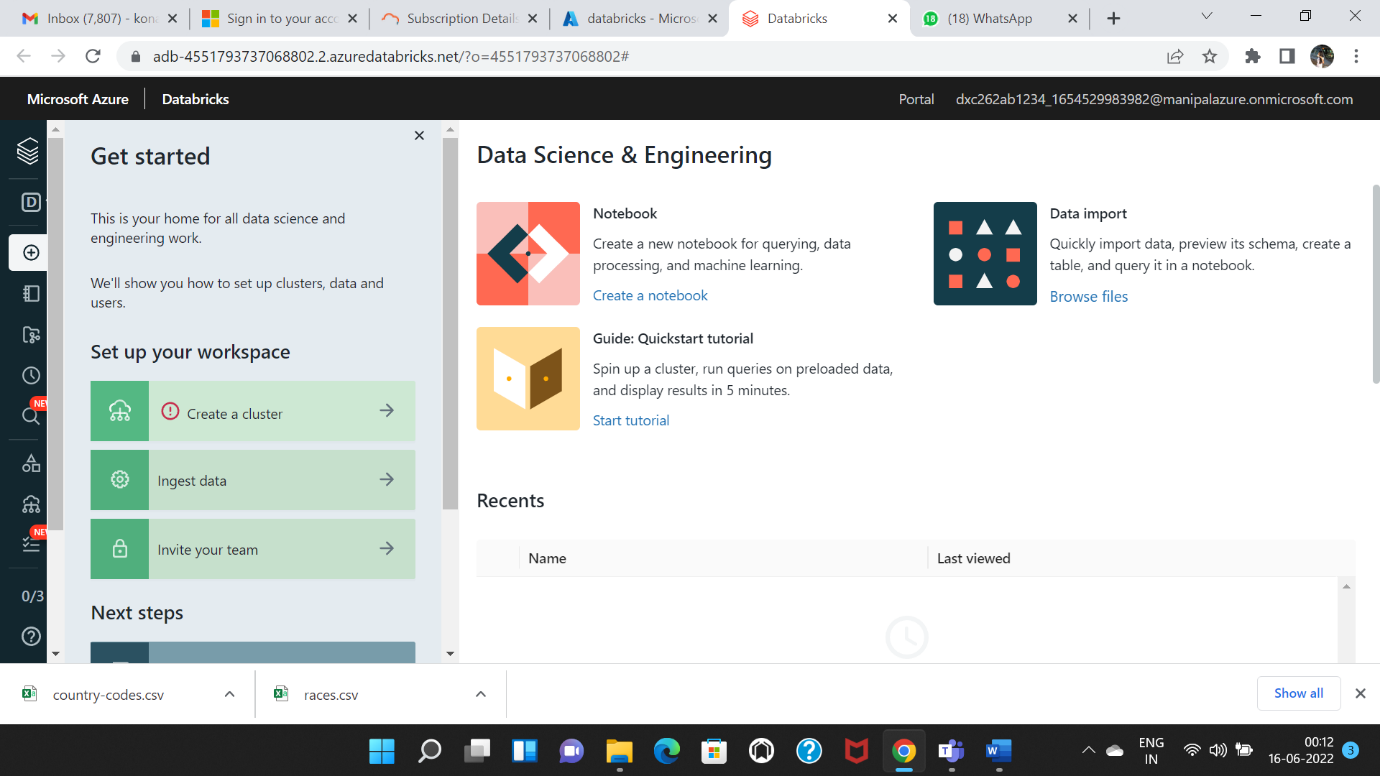
3. Using archive3.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

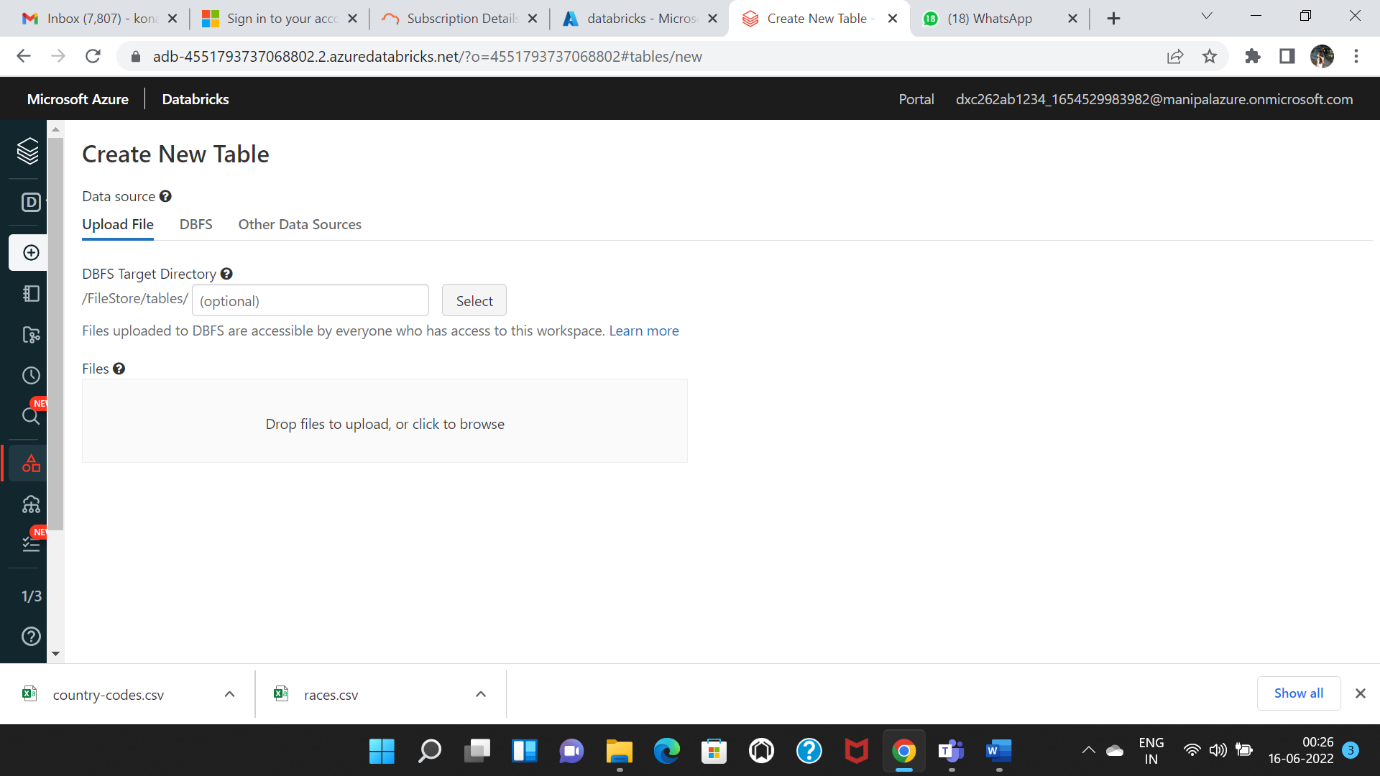
First, login to your Azure Portal and create a Databricks workspace.

Open the Databricks workspace and create cluster for your future use.

Now, create a notebook by clicking on the create Notebook option from the side panel.

After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the drag & drop region.





Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType

final\_data\_schema = StructType(fields=[StructField("tweet\_text",StringType(),False),

StructField("emotion\_in\_tweet\_is\_directed\_at",StringType(),True),

StructField("is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product",StringType(),True),

])

final\_data\_df = spark.read \

.option("header" , True) \

.schema(nces330\_20\_schema) \

.csv("/FileStore/tables/final\_data.csv")

A screenshot of a computer

Description automatically generated

from pyspark.sql.functions import col,lit

final\_data\_selected\_df = final\_data\_df.select(col('tweet\_text'),

col('emotion\_in\_tweet\_is\_directed\_at').alias('emotion\_towards'),col('is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product').alias('is\_there\_a\_brand'))

Graphical user interface, text, application, email

Description automatically generated

display(final\_data\_selected\_df)

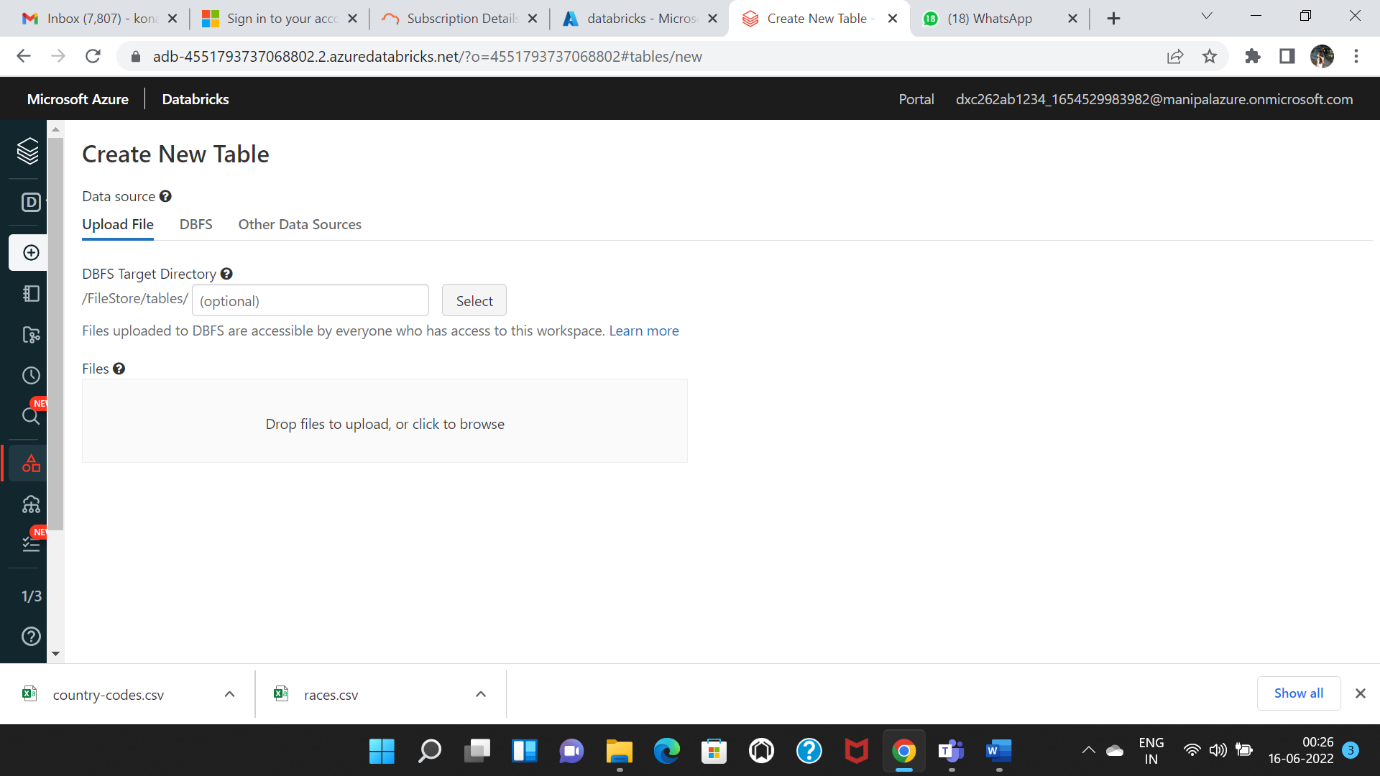
4. Using archive4.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

First, login to your Azure Portal and create a Databricks workspace.

Open the Databricks workspace and create cluster for your future use.

Now, create a notebook by clicking on the create Notebook option from the side panel.

After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the drag & drop region.



Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType

from pyspark.sql.types import StructType, StructField, IntegerType, StringType

SEntFiN-v1­\_1\_schema = StructType(fields=[StructField("S No.",IntegerType(),False),

StructField("Title",StringType(),True),

StructField("Decisions",StringType(),True),

StructField("Words",IntegerType(),True),

])

SEntFiN-v1\_1\_df = spark.read \

.option("header" , True) \

.schema(SEntFiN-v1\_1\_schema) \

.csv("/FileStore/tables/ SEntFiN-v1\_1.csv")

from pyspark.sql.functions import col,lit

SEntFiN-v1\_1\_selected\_df = SEntFiN-v1\_1\_df.select(col('S No'),col('Title'),col('Words’))

display(SEntFiN-v1\_1\_df)

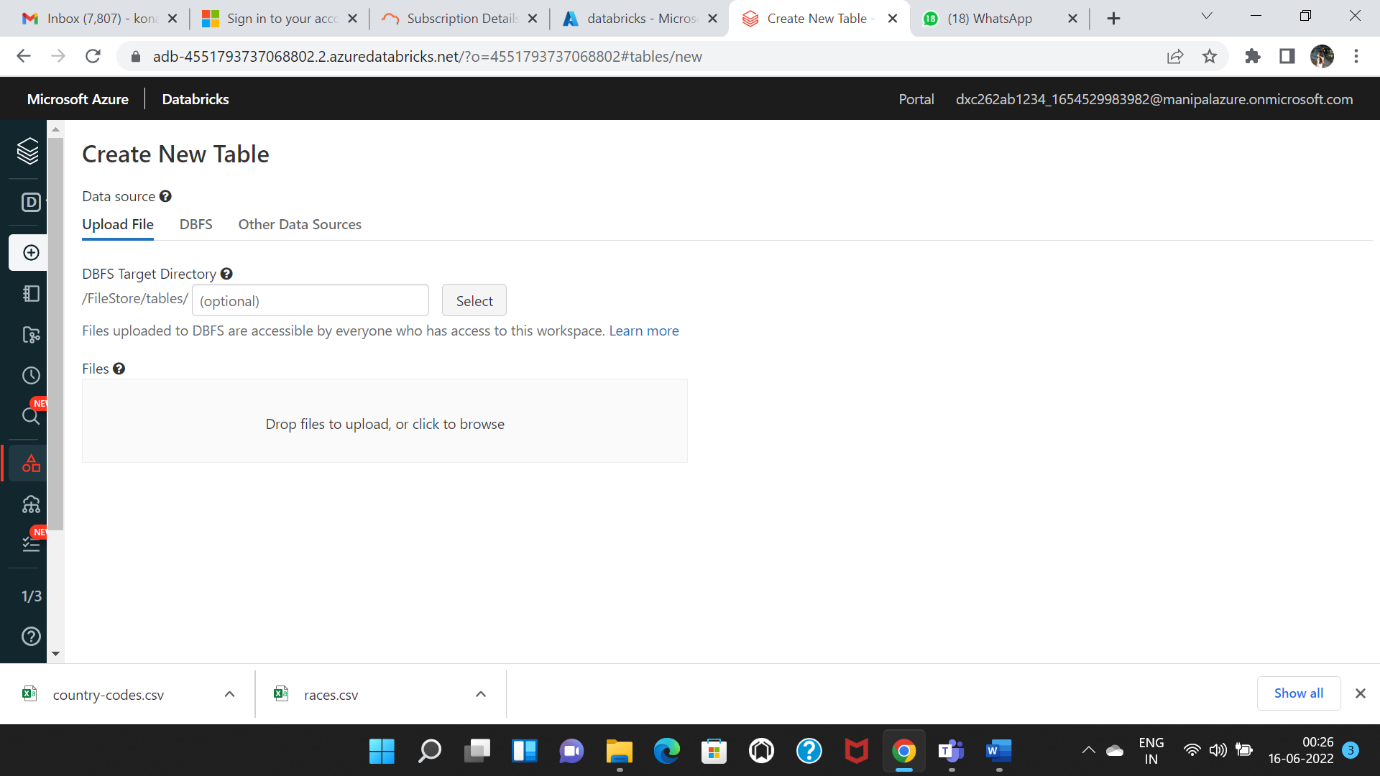
5. Using archive5.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

First, login to your Azure Portal and create a Databricks workspace.

Open the Databricks workspace and create cluster for your future use.

Now, create a notebook by clicking on the create Notebook option from the side panel.

After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the drag & drop region.



Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, FloatType

cancer\_death\_rates\_schema = StructType(fields=[StructField("Entity",StringType(),False),

StructField("Code",StringType(),True),

StructField("Year",IntegerType(),True),

StructField("Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate)",FloatType(),True),

])

Cancer\_death\_rates\_df = spark.read \

.option("header" , True) \

.schema(cancer\_death\_rates\_schema) \

.csv("/FileStore/tables/ cancer\_death\_rates.csv")

from pyspark.sql.functions import col,lit

cancer\_death\_rates\_selected\_df = cancer\_death\_rates\_df.select(col(' Entity'),

col(' Year'),col(' Deaths - Neoplasms - Sex: Both - Age: Age-standardized (Rate)’).alias(‘Deaths’))

display(cancer\_death\_rates\_df)

6. Using archive6.zip file - please ingest data into Databricks DBFS path & query the data, redesign columns accordingly using Dataframe commands - display with notebooks accordingly.

First, login to your Azure Portal and create a Databricks workspace.

Open the Databricks workspace and create cluster for your future use.

Now, create a notebook by clicking on the create Notebook option from the side panel.

After creating the notebook, ingest the data into the Databricks by dragging and dropping the required file in the drag & drop region.

Import the required fields and features from pyspark.

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, FloatType

inflation\_gdp\_schema = StructType(fields=[StructField("Country ",StringType(),False),

StructField("Country Code ",StringType(),True),

StructField("Year ",IntegerType(),True),

StructField("Inflation ",FloatType(),True),

])

inflation\_gdp \_df = spark.read \

.option("header" , True) \

.schema(inflation\_gdp \_schema) \

.csv("/FileStore/tables/ inflation\_gdp.csv")

from pyspark.sql.functions import col,lit

inflation\_gdp \_selected\_df = inflation\_gdp \_df.select(col('S No'),

col('Title'),col('Words’))

display(inflation\_gdp \_df)