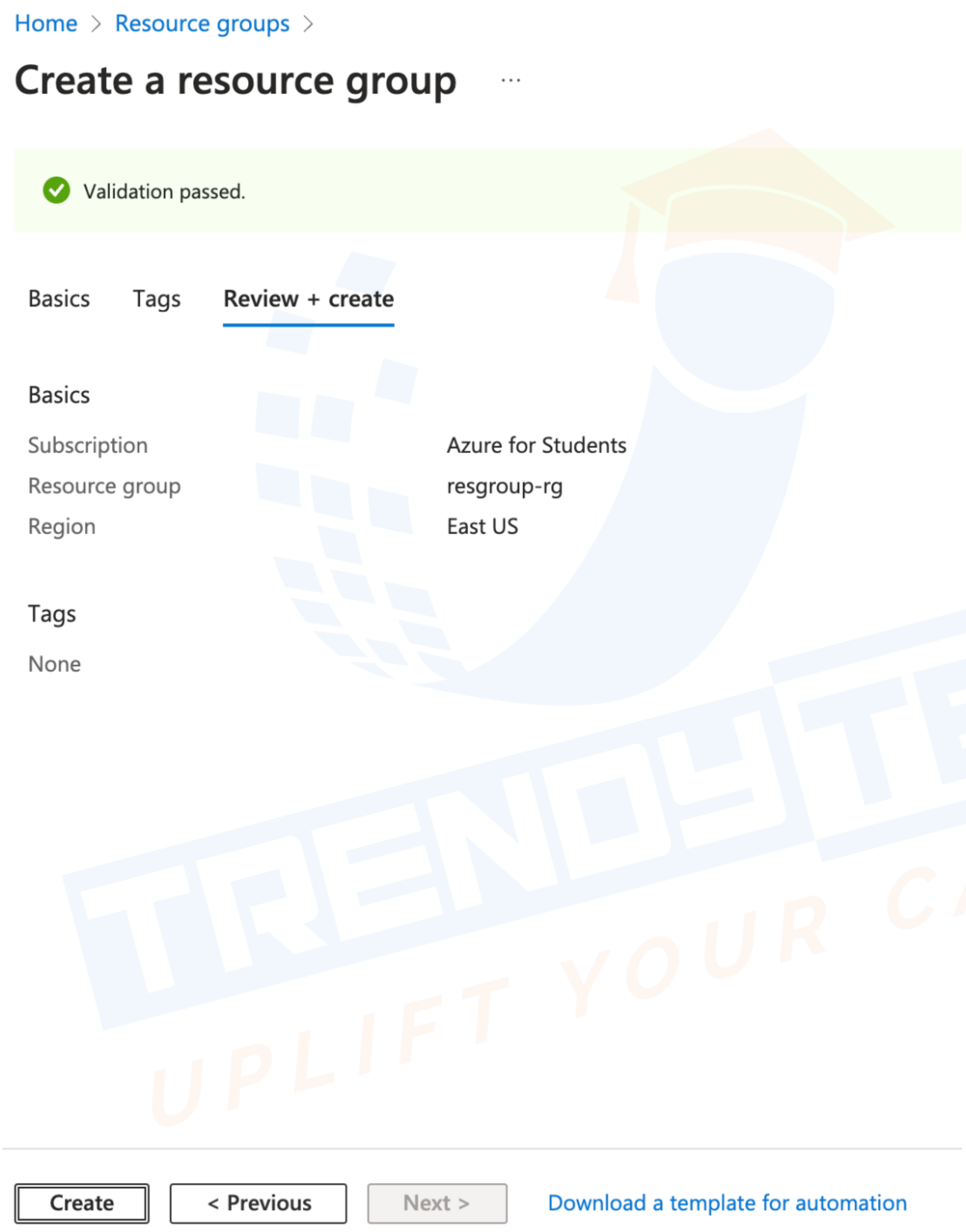
Answer 1:

- Search resource group
- Press- +create
- Give name for the resource group
- Press- review + create
- And create



Create a resource group

Validation passed.

Basics

Subscription	Azure for Students
Resource group	resgroup-rg
Region	East US

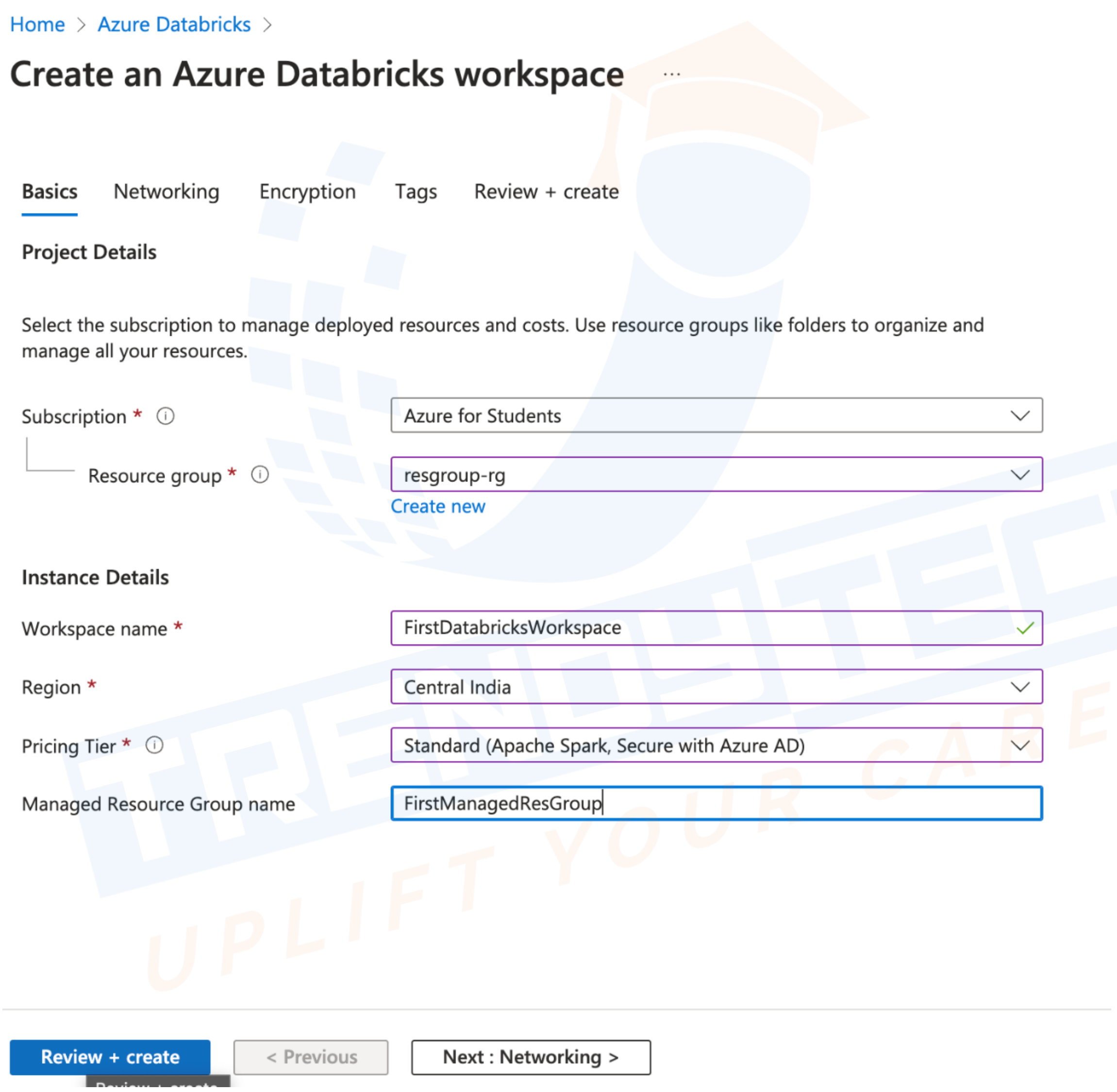
Tags

None

Answer 2

Answer 2a

- Search azure data bricks
- Select subscription
- Select resource group
- give workspace name
- select region, tier
- give managed resource group name



Create an Azure Databricks workspace

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

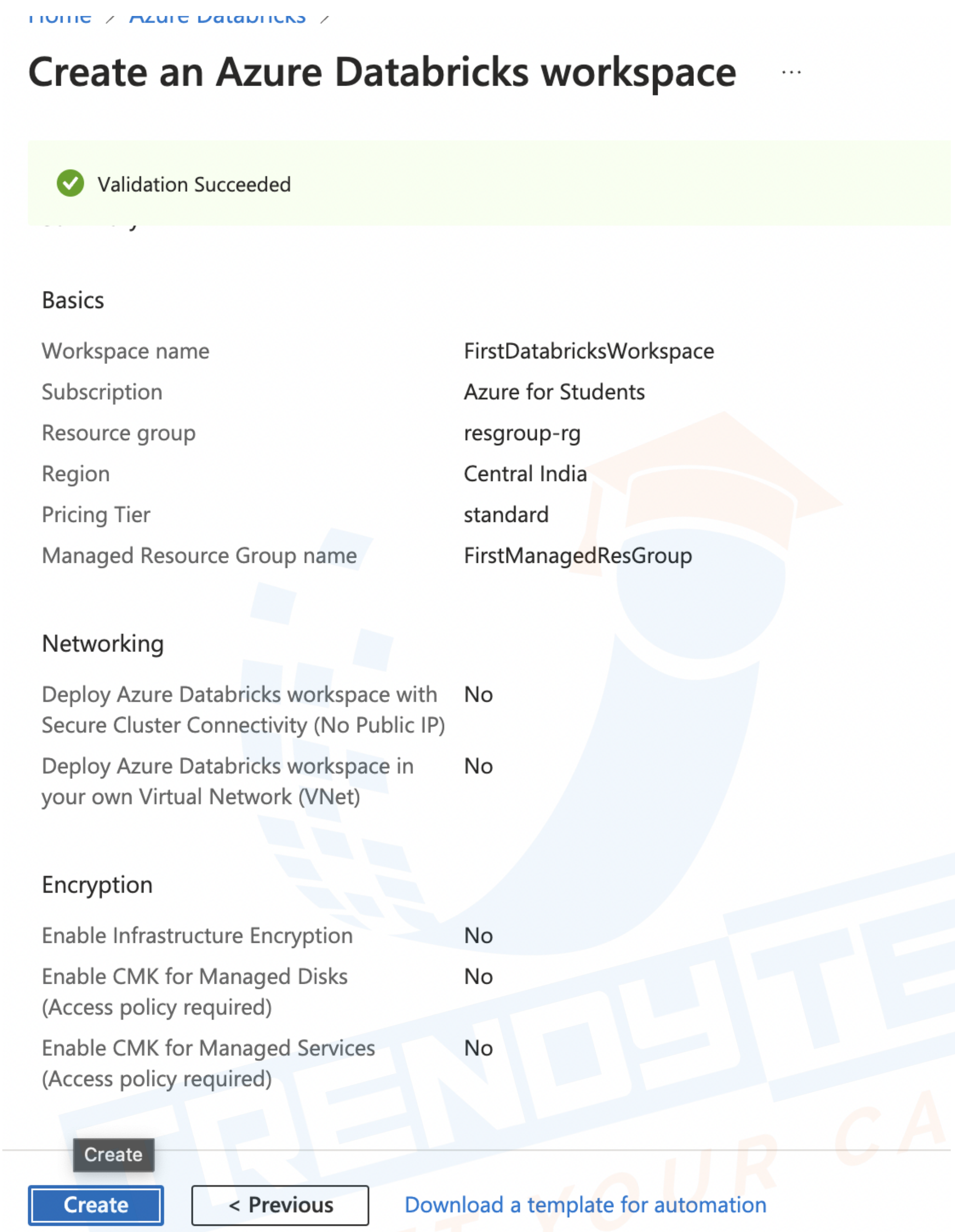
Subscription *	Azure for Students
Resource group *	resgroup-rg

Create new

Instance Details

Workspace name *	FirstDatabricksWorkspace
Region *	Central India
Pricing Tier *	Standard (Apache Spark, Secure with Azure AD)
Managed Resource Group name	FirstManagedResGroup

- Press Review + create
- And create



Home / Azure Databricks /

Create an Azure Databricks workspace

✓ Validation Succeeded

Basics

Workspace name	FirstDatabricksWorkspace
Subscription	Azure for Students
Resource group	resgroup-rg
Region	Central India
Pricing Tier	standard
Managed Resource Group name	FirstManagedResGroup

Networking

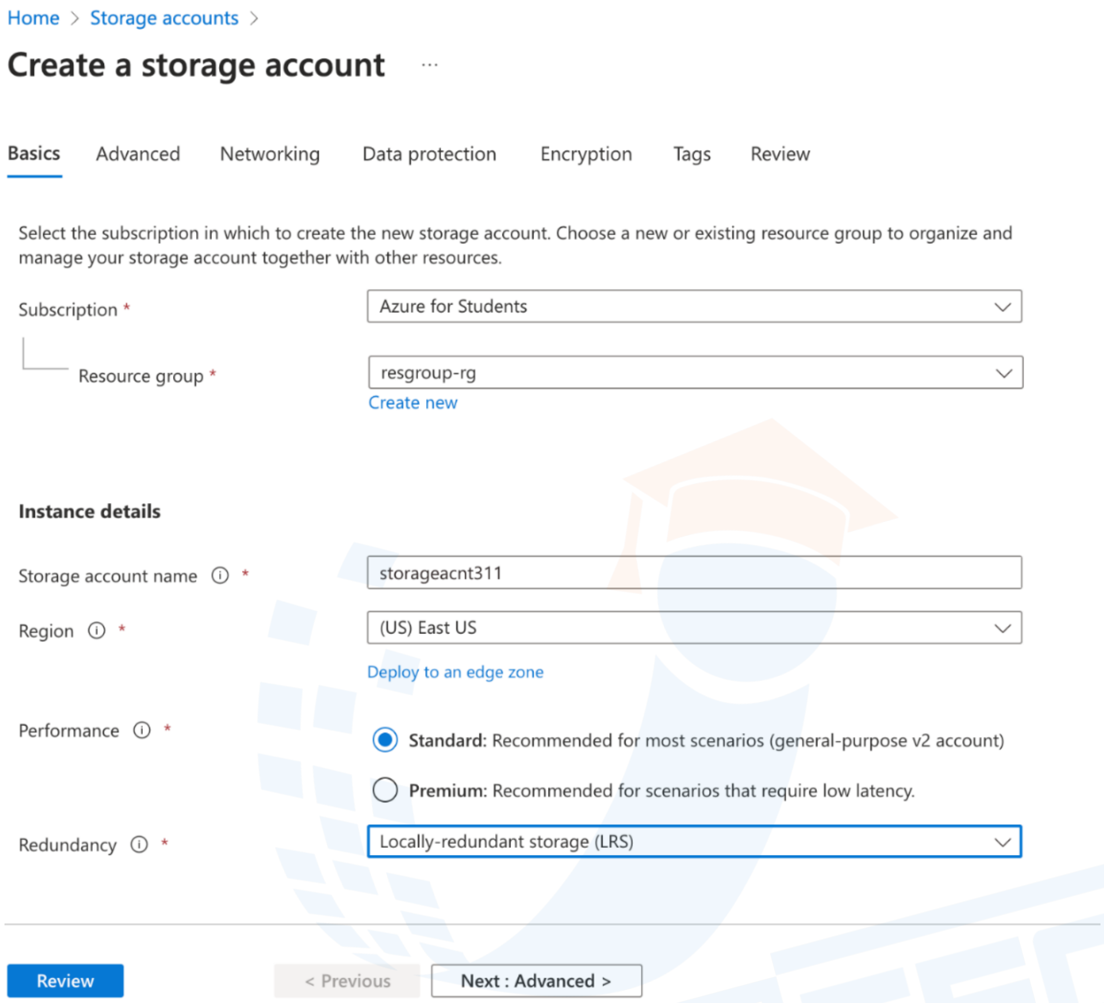
Deploy Azure Databricks workspace with Secure Cluster Connectivity (No Public IP)	No
Deploy Azure Databricks workspace in your own Virtual Network (VNet)	No

Encryption

Enable Infrastructure Encryption	No
Enable CMK for Managed Disks (Access policy required)	No
Enable CMK for Managed Services (Access policy required)	No

Answer 2b

- Search storage accounts
- Create storage account
- select resource group
- give any storage account name



Create a storage account

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *: Azure for Students

Resource group *: resgroup-rg

Instance details

Storage account name ⓘ *: storageacnt311

Region ⓘ *: (US) East US

Performance ⓘ *

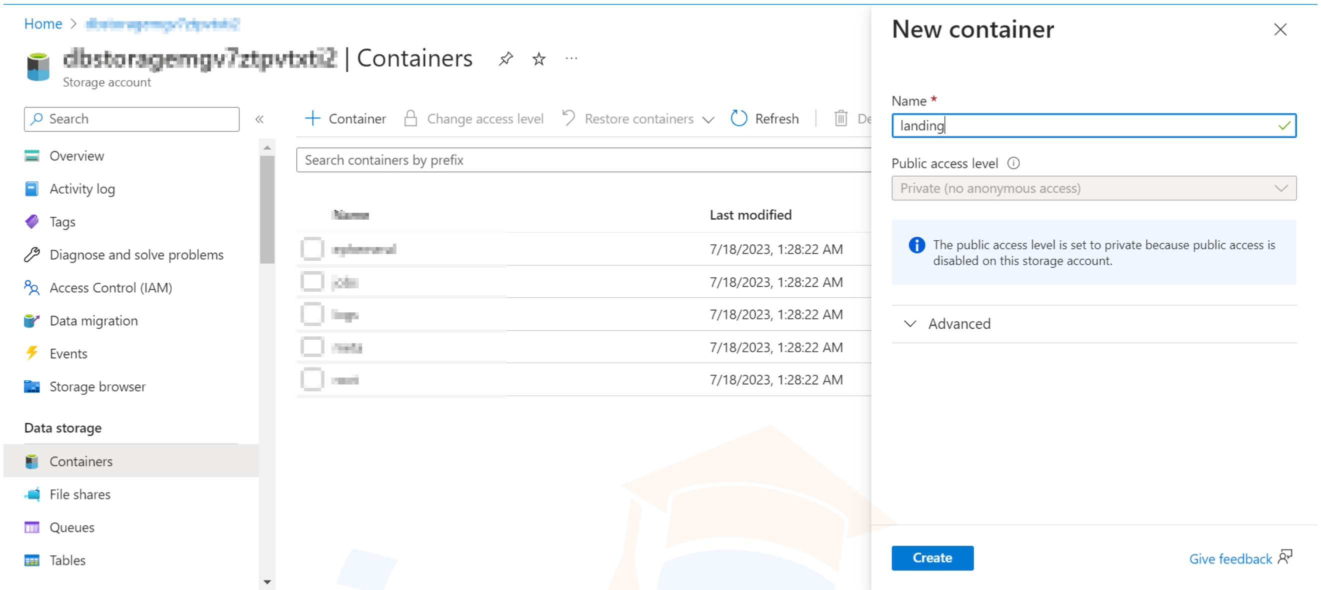
● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ *: Locally-redundant storage (LRS)

Answer 2c

- You may use ADLS Gen2 or Blob Store, it’s your choice. Usually for structured data/files, we prefer ADLS Gen2.
- Creating a landing container.



[redacted]	Last modified
□ [redacted]	7/18/2023, 1:28:22 AM
□ [redacted]	7/18/2023, 1:28:22 AM
□ [redacted]	7/18/2023, 1:28:22 AM
□ [redacted]	7/18/2023, 1:28:22 AM
□ [redacted]	7/18/2023, 1:28:22 AM

New container

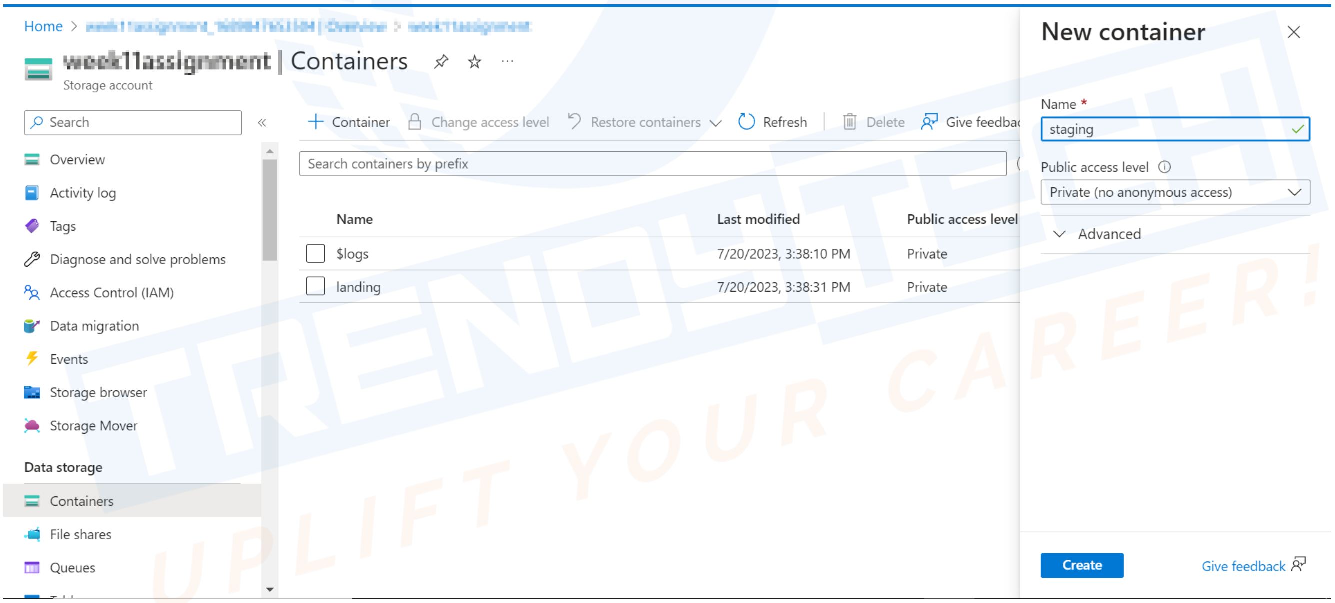
Name * landing

Public access level: Private (no anonymous access)

The public access level is set to private because public access is disabled on this storage account.

Advanced

- Creating a staging container.



Name	Last modified	Public access level
□ $logs	7/20/2023, 3:38:10 PM	Private
□ landing	7/20/2023, 3:38:31 PM	Private

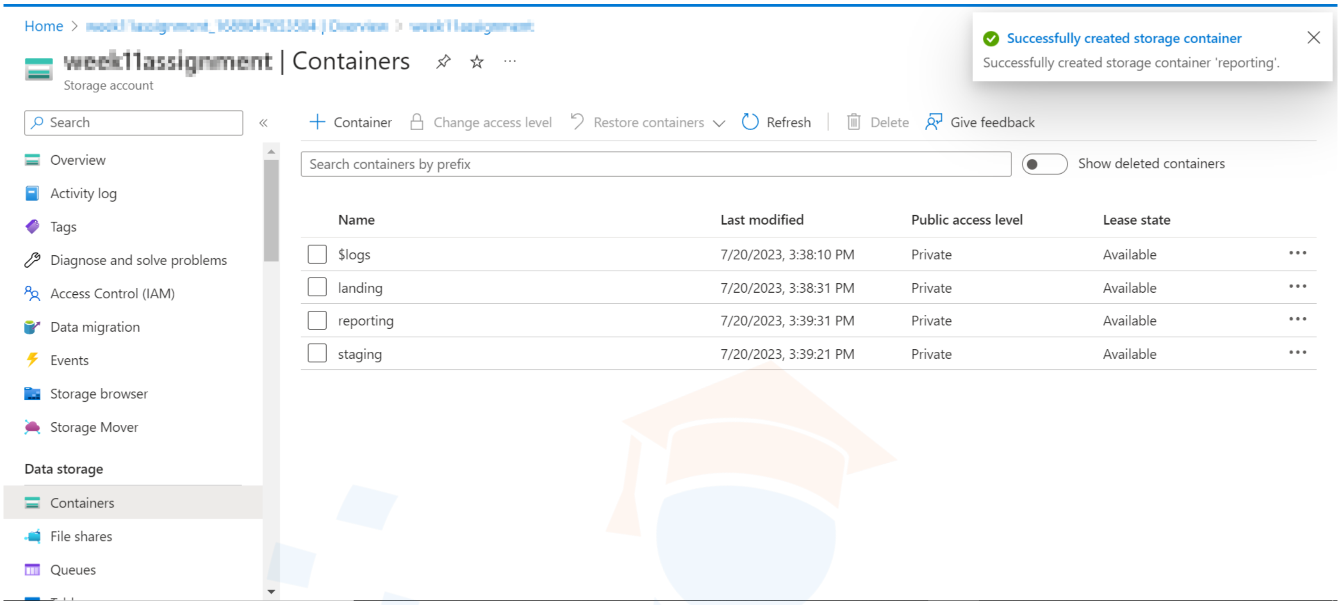
New container

Name * staging

Public access level: Private (no anonymous access)

Advanced

- Creating a reporting container.



Home > [redacted] > [redacted]

[redacted] | Containers

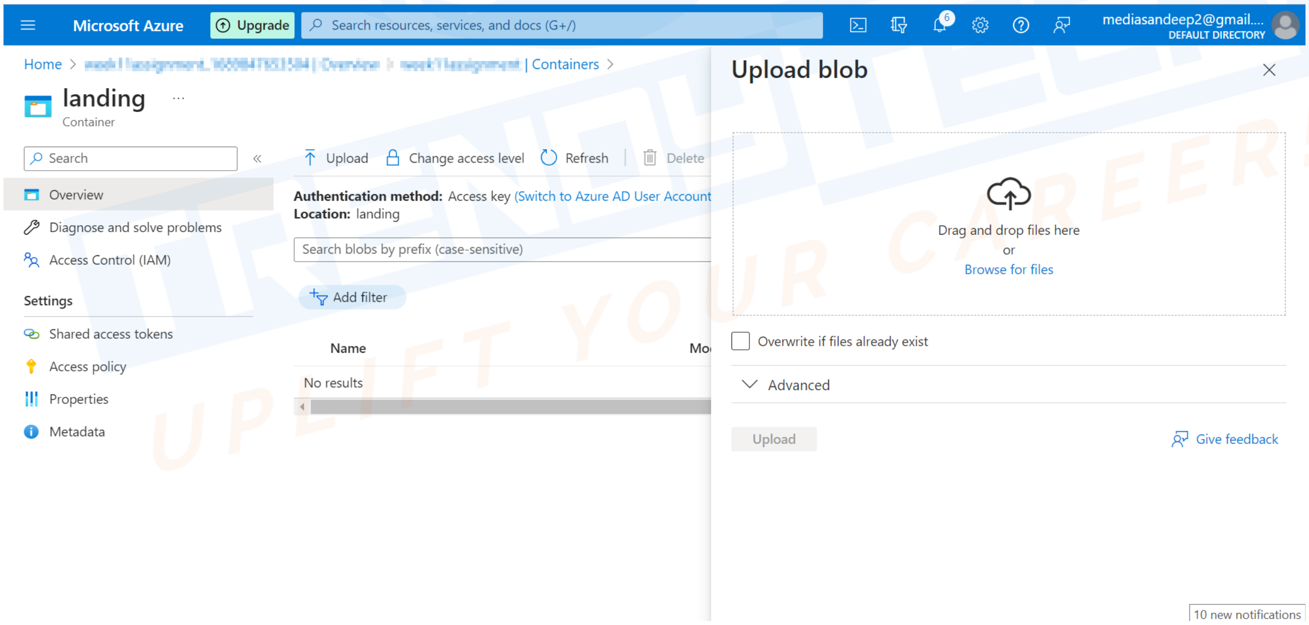
Storage account

Successfully created storage container

Successfully created storage container 'reporting'.

Name	Last modified	Public access level	Lease state
$logs	7/20/2023, 3:38:10 PM	Private	Available
landing	7/20/2023, 3:38:31 PM	Private	Available
reporting	7/20/2023, 3:39:31 PM	Private	Available
staging	7/20/2023, 3:39:21 PM	Private	Available

- Adding customer and orders dataset in landing container.



Home > [redacted] > [redacted] | Containers >

landing

Container

Authentication method: Access key (Switch to Azure AD User Account

Location: landing

Name | Mo

No results

Upload blob

Drag and drop files here

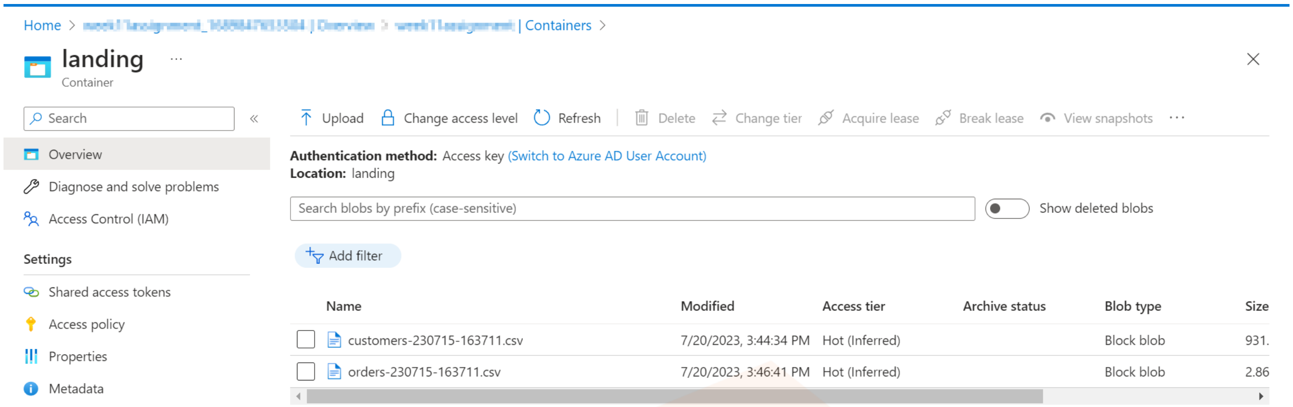
or

Browse for files

□ Overwrite if files already exist

Advanced

- Orders and Customers file in the landing container.

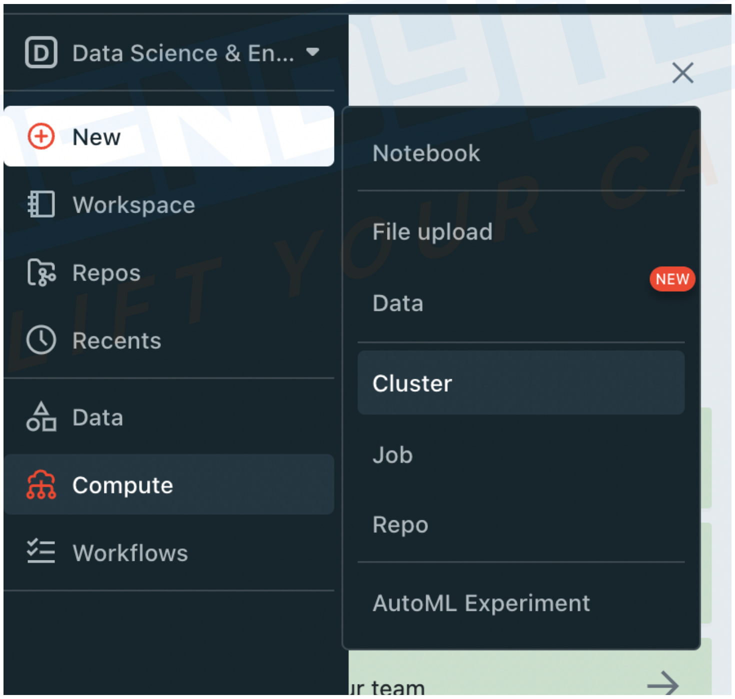


Authentication method: Access key (Switch to Azure AD User Account)
Location: landing

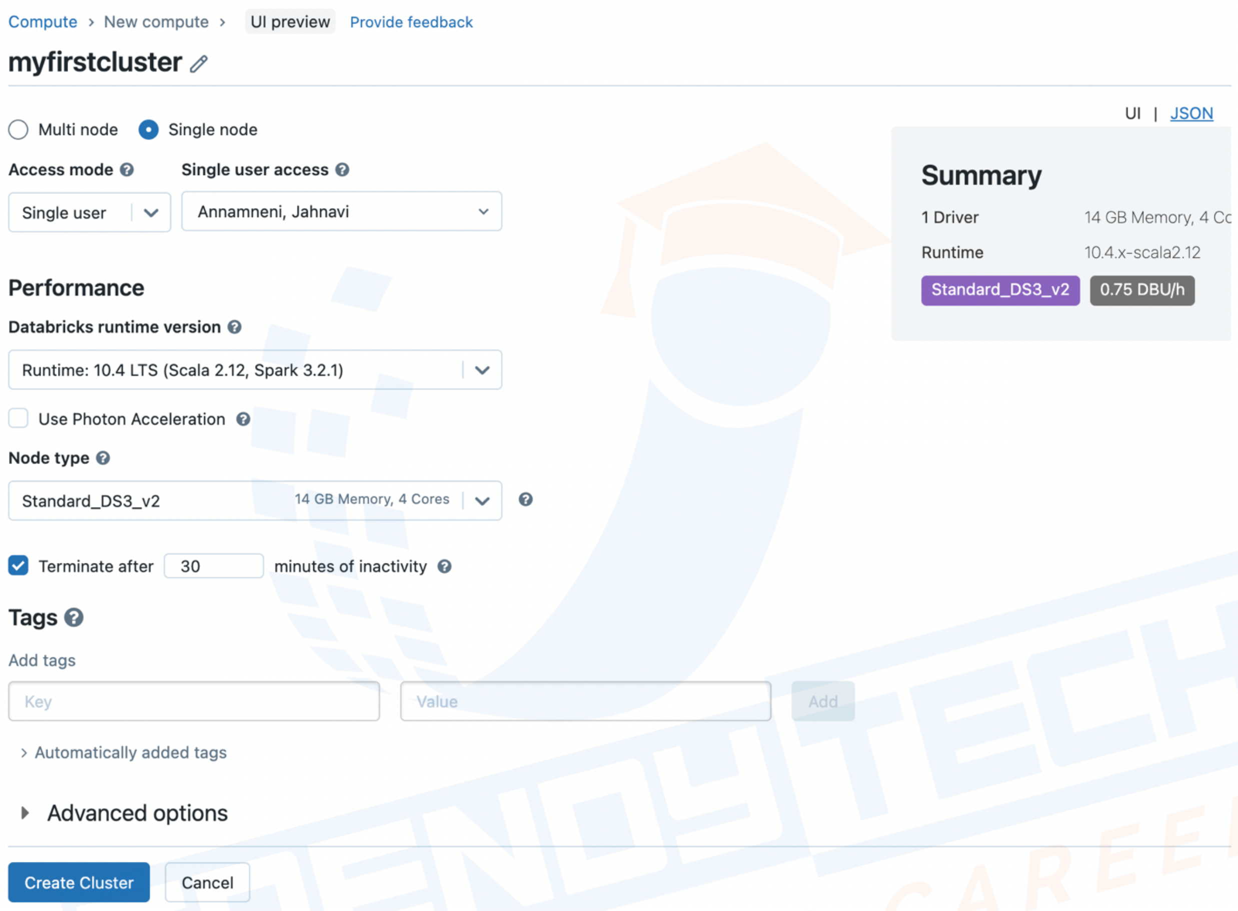
Name	Modified	Access tier	Archive status	Blob type	Size
□ customers-230715-163711.csv	7/20/2023, 3:44:34 PM	Hot (Inferred)		Block blob	931.
□ orders-230715-163711.csv	7/20/2023, 3:46:41 PM	Hot (Inferred)		Block blob	2.86

Answer 3

- Go to All resources
- Select the azure databricks workspace created
- Enter launch workspace
- Select DataScience and Engineering
- Creating the cluster
    - o Select new -> cluster



- Give cluster name
- Select single node -> version -> node type as mentioned in question -> terminate after 30 mins of inactivity



Compute › New compute › UI preview Provide feedback

myfirstcluster

○ Multi node ● Single node

Access mode: Single user

Single user access: Annamneni, Jahnavi

Performance

Databricks runtime version: Runtime: 10.4 LTS (Scala 2.12, Spark 3.2.1)

□ Use Photon Acceleration

Node type: Standard_DS3_v2 14 GB Memory, 4 Cores

☑ Terminate after 30 minutes of inactivity

Tags

Add tags

Key | Value | Add

› Automatically added tags

Advanced options

Create Cluster | Cancel

UI | JSON

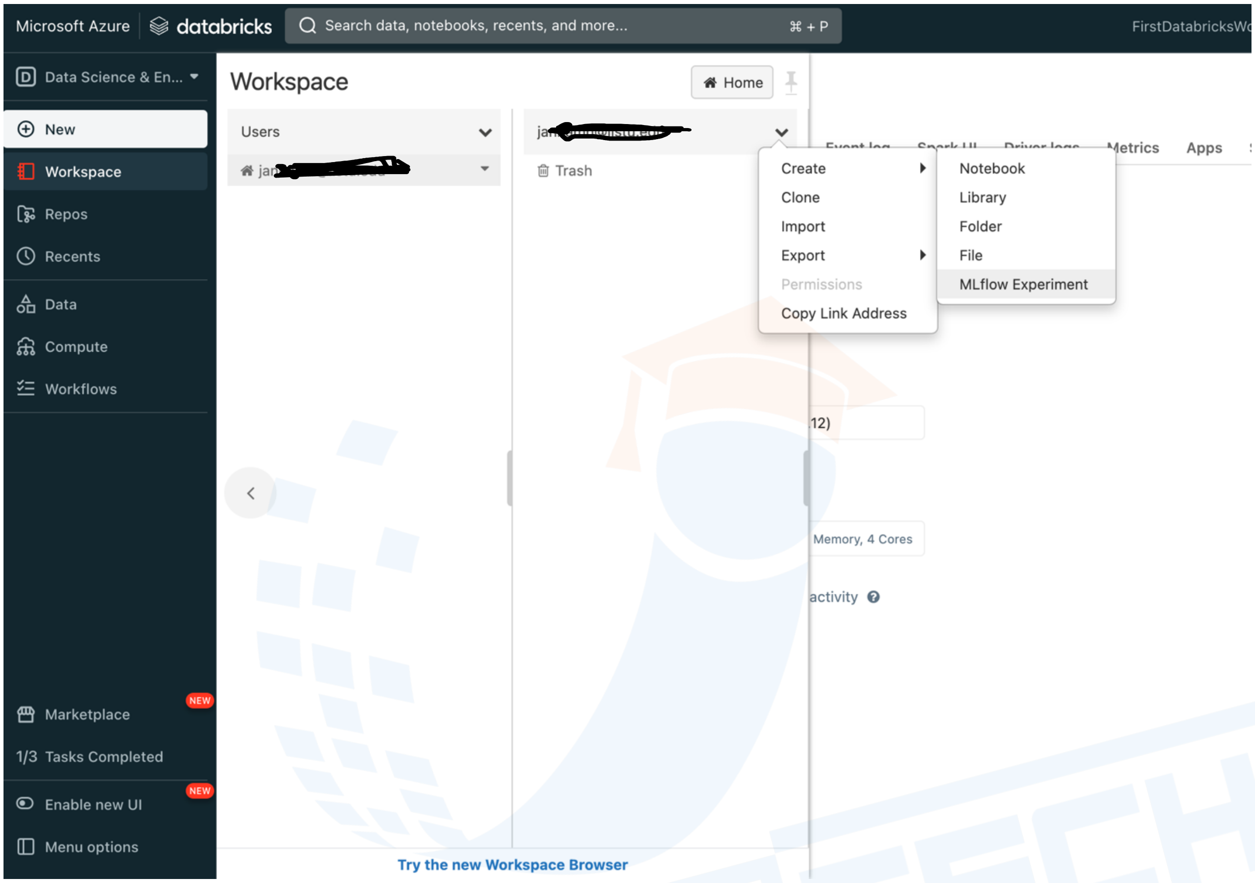
Summary

1 Driver	14 GB Memory, 4 Co
Runtime	10.4.x-scala2.12

Standard_DS3_v2 | 0.75 DBU/h

Answer 4

- Go to new -> user -> create -> notebook



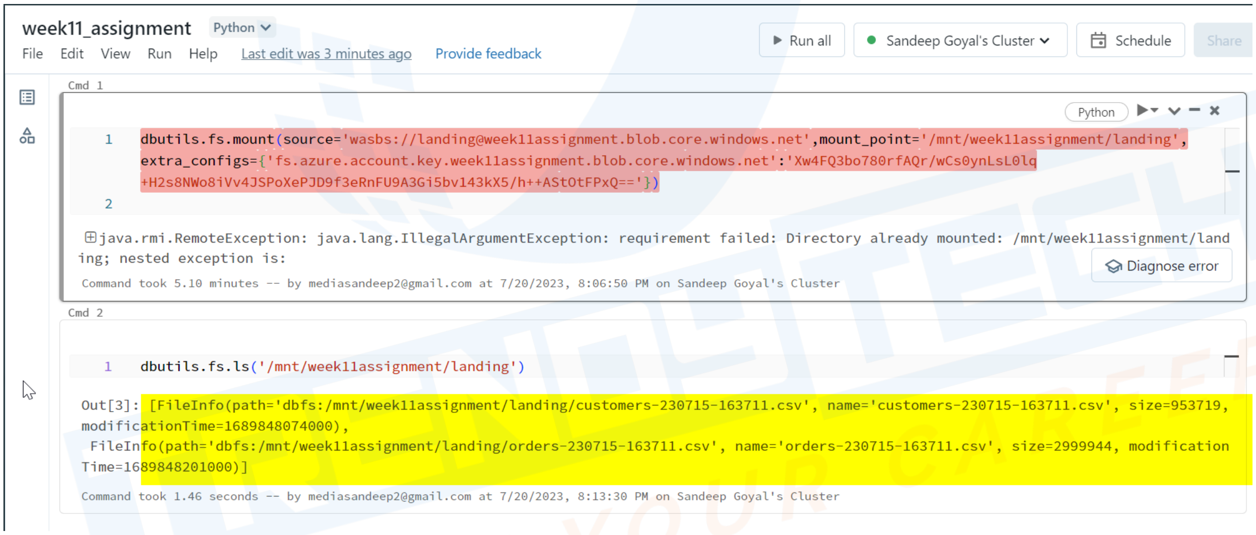
Answer 4a

- For mounting, we would need to generate an access key.
  - For the key-
  - go to storage account -> access keys- copy one key and paste it.

```
#Ans 4a - mounting the landing container
dbutils.fs.mount(source='wasbs://landing@week11assignment.blob.core.windows.net',mount_point='/mnt/week11assignment/landing',extra_configs={'fs.azure.account.key.week11assignment.blob.core.windows.net':'Xw4FQ3bo780rfAQr/wCs0ynLsL0lq+H2s8NWo8iVv4JSPoXePJD9f3eRnFU9A3Gi5bv143kX5/h++AStOtFPxQ=='})

#Ans 4a - validating the files in the landing container
dbutils.fs.ls('/mnt/week11assignment/landing')
```

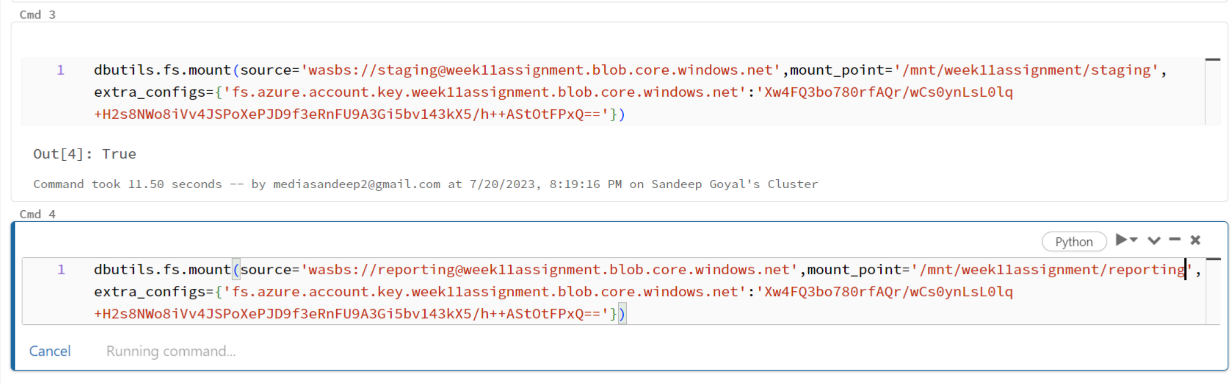
```
#Ans 4a - mounting the staging container
dbutils.fs.mount(source='wasbs://staging@week11assignment.blob.core.windows.net',mount_point='/mnt/week11assignment/staging',extra_configs={'fs.azure.account.key.week11assignment.blob.core.windows.net':'Xw4FQ3bo780rfAQr/wCs0ynLsL0lq+H2s8NWo8iVv4JSPoXePJD9f3eRnFU9A3Gi5bv143kX5/h++AStOtFPxQ=='})


#Ans 4a - mounting the reporting container
dbutils.fs.mount(source='wasbs://reporting@week11assignment.blob.core.windows.net',mount_point='/mnt/week11assignment/reporting',extra_configs={'fs.azure.account.key.week11assignment.blob.core.windows.net':'Xw4FQ3bo780rfAQr/wCs0ynLsL0lq+H2s8NWo8iVv4JSPoXePJD9f3eRnFU9A3Gi5bv143kX5/h++AStOtFPxQ=='})
```

- Code Snippet



```
dbutils.fs.mount(source='wasbs://landing@week11assignment.blob.core.windows.net',mount_point='/mnt/week11assignment/landing',
extra_configs={'fs.azure.account.key.week11assignment.blob.core.windows.net':'Xw4FQ3bo780rfAQr/wCs0ynLsL0lq
+H2s8NWo8iVv4JSPoXePJD9f3eRnFU9A3Gi5bv143kX5/h++AStOtFPxQ=='})

⊞java.rmi.RemoteException: java.lang.IllegalArgumentException: requirement failed: Directory already mounted: /mnt/week11assignment/land
ing; nested exception is:
Command took 5.10 minutes -- by mediasandeep2@gmail.com at 7/20/2023, 8:06:50 PM on Sandeep Goyal's Cluster
```

```
dbutils.fs.ls('/mnt/week11assignment/landing')

Out[3]: [FileInfo(path='dbfs:/mnt/week11assignment/landing/customers-230715-163711.csv', name='customers-230715-163711.csv', size=953719,
modificationTime=1689848074000),
 FileInfo(path='dbfs:/mnt/week11assignment/landing/orders-230715-163711.csv', name='orders-230715-163711.csv', size=2999944, modification
Time=1689848201000)]
Command took 1.46 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 8:13:30 PM on Sandeep Goyal's Cluster
```



Cmd 3

```
dbutils.fs.mount(source='wasbs://staging@week11assignment.blob.core.windows.net',mount_point='/mnt/week11assignment/staging',
extra_configs={'fs.azure.account.key.week11assignment.blob.core.windows.net':'Xw4FQ3bo780rfAQr/wCs0ynLsL0lq
+H2s8NWo8iVv4JSPoXePJD9f3eRnFU9A3Gi5bv143kX5/h++AStOtFPxQ=='})
```

```
Out[4]: True
```

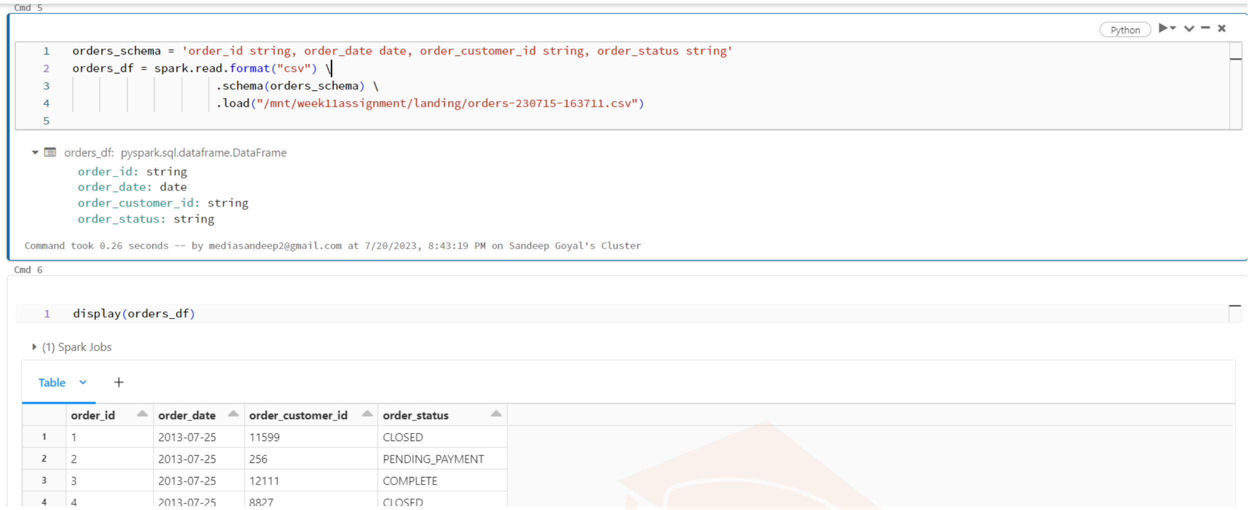
Command took 11.50 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 8:19:16 PM on Sandeep Goyal's Cluster

Cmd 4

```
dbutils.fs.mount(source='wasbs://reporting@week11assignment.blob.core.windows.net',mount_point='/mnt/week11assignment/reporting',
extra_configs={'fs.azure.account.key.week11assignment.blob.core.windows.net':'Xw4FQ3bo780rfAQr/wCs0ynLsL0lq
+H2s8NWo8iVv4JSPoXePJD9f3eRnFU9A3Gi5bv143kX5/h++AStOtFPxQ=='})
```

Answer 4b

```
#Ans 4b - reading the file from landing folder
orders_schema = 'order_id string, order_date date, order_customer_id string,
order_status string'
orders_df = spark.read.format("csv") \
                    .schema(orders_schema) \

 .load("/mnt/week11assignment/landing/orders-230715-163711.csv")

#Ans 4b - display the orders_df
display(orders_df)

#Ans 4b - reading the customers file
customers_schema = 'customer_id string, customer_fname string,
customer_lname string, customer_email string, customer_password string,
customer_street string, customer_city string, customer_state string,
customer_zipcode string'
customers_df = spark.read.format("csv") \
                    .schema(customers_schema) \

 .load("/mnt/week11assignment/landing/customers-230715-163711.csv")

#Ans 4b - display the customers_df
display(customers_df)
```



Cmd 5

```python
orders_schema = 'order_id string, order_date date, order_customer_id string, order_status string'
orders_df = spark.read.format("csv") \
                    .schema(orders_schema) \
                    .load("/mnt/week11assignment/landing/orders-230715-163711.csv")
```

```
orders_df:  pyspark.sql.dataframe.DataFrame
    order_id: string
    order_date: date
    order_customer_id: string
    order_status: string
```

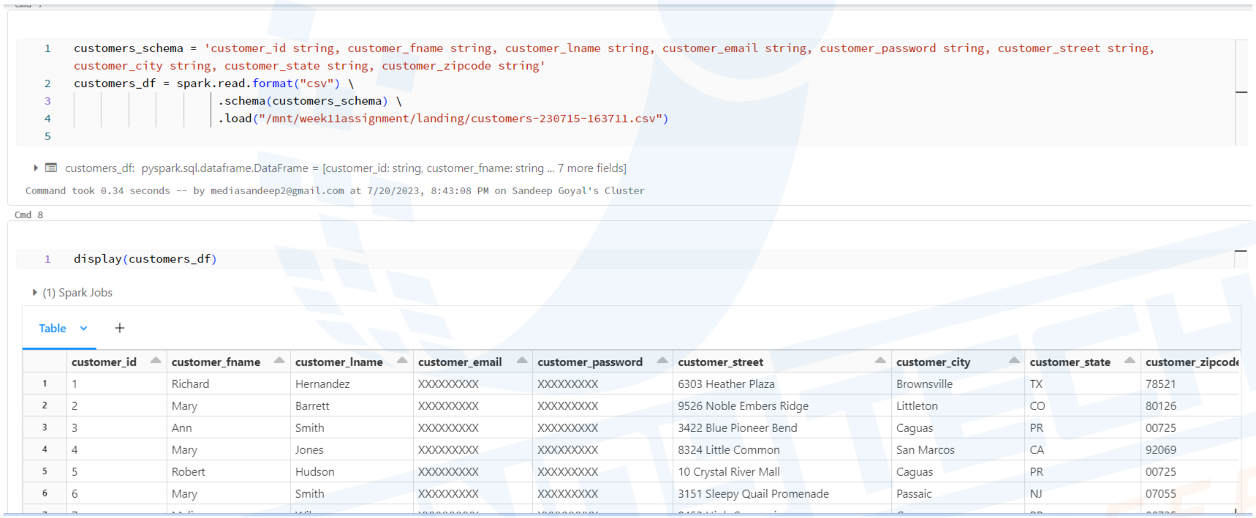
```
Command took 0.26 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 8:43:19 PM on Sandeep Goyal's Cluster
```

Cmd 6

```python
display(orders_df)
```

(1) Spark Jobs

	order_id	order_date	order_customer_id	order_status
1	1	2013-07-25	11599	CLOSED
2	2	2013-07-25	256	PENDING_PAYMENT
3	3	2013-07-25	12111	COMPLETE
4	4	2013-07-25	8827	CLOSED



```python
customers_schema = 'customer_id string, customer_fname string, customer_lname string, customer_email string, customer_password string, customer_street string, customer_city string, customer_state string, customer_zipcode string'
customers_df = spark.read.format("csv") \
                    .schema(customers_schema) \
                    .load("/mnt/week11assignment/landing/customers-230715-163711.csv")
```

```
customers_df:  pyspark.sql.dataframe.DataFrame = [customer_id: string, customer_fname: string ... 7 more fields]
Command took 0.34 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 8:43:08 PM on Sandeep Goyal's Cluster
```

Cmd 8

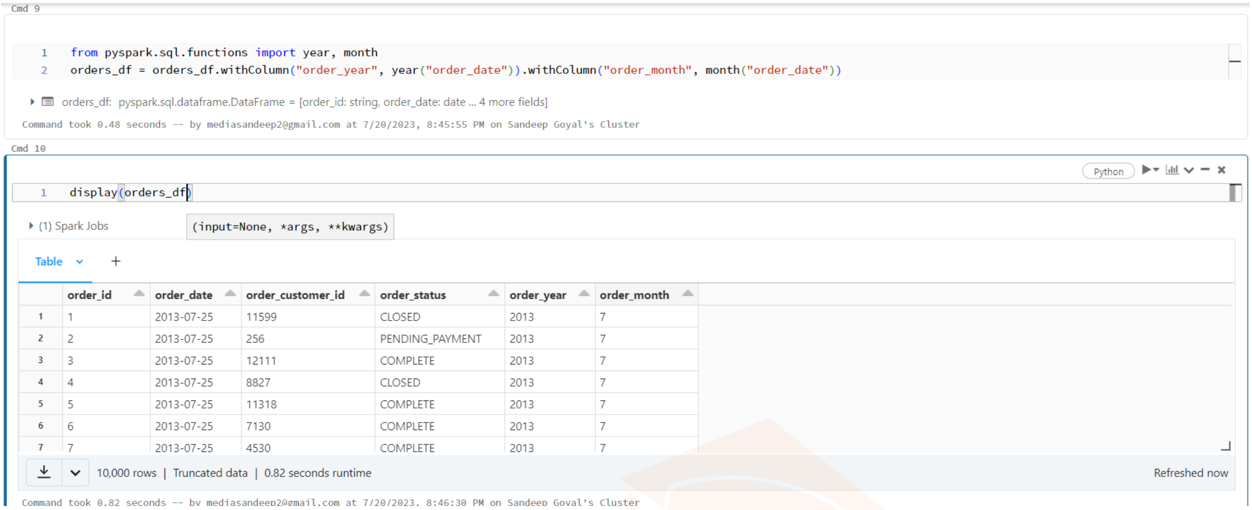
```python
display(customers_df)
```

(1) Spark Jobs

	customer_id	customer_fname	customer_lname	customer_email	customer_password	customer_street	customer_city	customer_state	customer_zipcode
1	1	Richard	Hernandez	XXXXXXXXX	XXXXXXXXX	6303 Heather Plaza	Brownsville	TX	78521
2	2	Mary	Barrett	XXXXXXXXX	XXXXXXXXX	9526 Noble Embers Ridge	Littleton	CO	80126
3	3	Ann	Smith	XXXXXXXXX	XXXXXXXXX	3422 Blue Pioneer Bend	Caguas	PR	00725
4	4	Mary	Jones	XXXXXXXXX	XXXXXXXXX	8324 Little Common	San Marcos	CA	92069
5	5	Robert	Hudson	XXXXXXXXX	XXXXXXXXX	10 Crystal River Mall	Caguas	PR	00725
6	6	Mary	Smith	XXXXXXXXX	XXXXXXXXX	3151 Sleepy Quail Promenade	Passaic	NJ	07055

Answer 4c

```python
#Ans 4c - transforming orders_df by adding columns to the order_df
from pyspark.sql.functions import year, month
orders_df = orders_df.withColumn("order_year",
year("order_date")).withColumn("order_month", month("order_date"))

#Ans 4c - display the transformed orders_df
display(orders_df)
```



Cmd 9

```
from pyspark.sql.functions import year, month
orders_df = orders_df.withColumn("order_year", year("order_date")).withColumn("order_month", month("order_date"))
```

```
orders_df: pyspark.sql.dataframe.DataFrame = [order_id: string, order_date: date ... 4 more fields]
Command took 0.48 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 8:45:55 PM on Sandeep Goyal's Cluster
```

Cmd 10

```
display(orders_df)
```

	order_id	order_date	order_customer_id	order_status	order_year	order_month
1	1	2013-07-25	11599	CLOSED	2013	7
2	2	2013-07-25	256	PENDING_PAYMENT	2013	7
3	3	2013-07-25	12111	COMPLETE	2013	7
4	4	2013-07-25	8827	CLOSED	2013	7
5	5	2013-07-25	11318	COMPLETE	2013	7
6	6	2013-07-25	7130	COMPLETE	2013	7
7	7	2013-07-25	4530	COMPLETE	2013	7

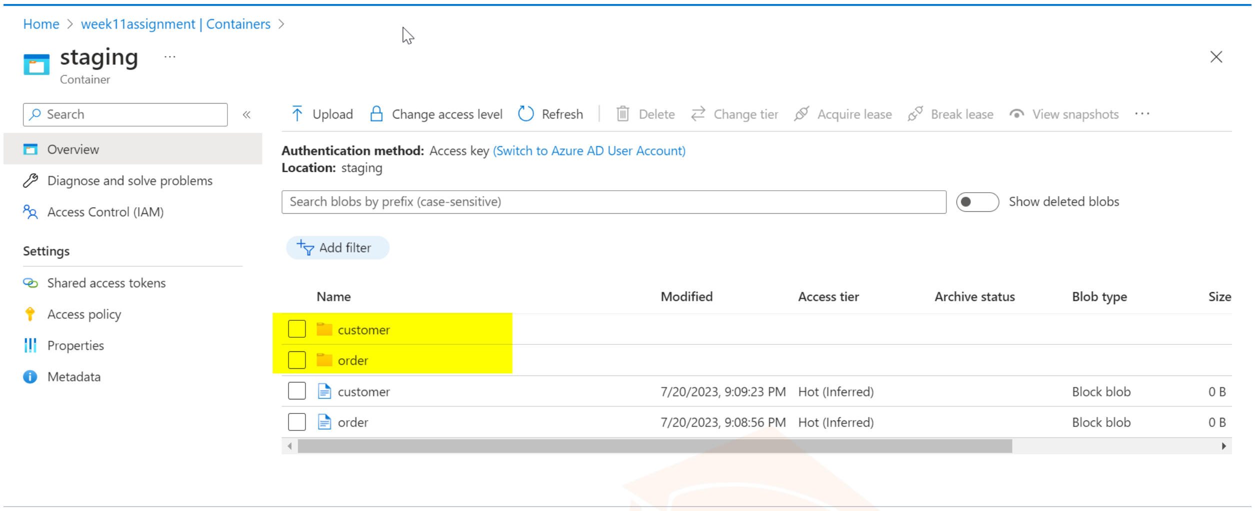
```
10,000 rows | Truncated data | 0.82 seconds runtime                Refreshed now
Command took 0.82 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 8:46:30 PM on Sandeep Goyal's Cluster
```

Answer 4d

- For optimal storage, we are using Parquet file format
- For better query performance per the query example provided, we have to partition on order_status and order_year in orders_df and customer_state in customers_df

```
#Ans 4d - writing the transformed orders_df to staging container
orders_df.write\
    .format("parquet")\
    .mode("overwrite")\
    .partitionBy("order_year","order_status")\
    .save("/mnt/week11assignment/staging/order")

#Ans 4d - writing the transformed customers_df to staging container
customers_df.write\
    .format("parquet")\
    .mode("overwrite")\
    .partitionBy("customer_state")\
    .save("/mnt/week11assignment/staging/customer")
```

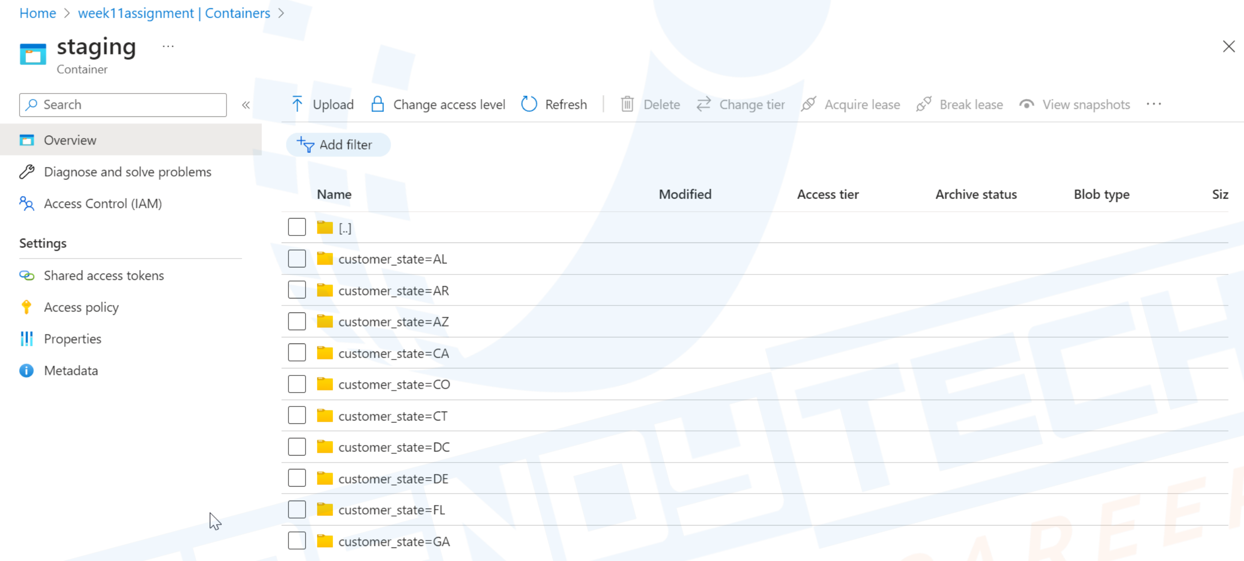
- *Staging Container Snippet*



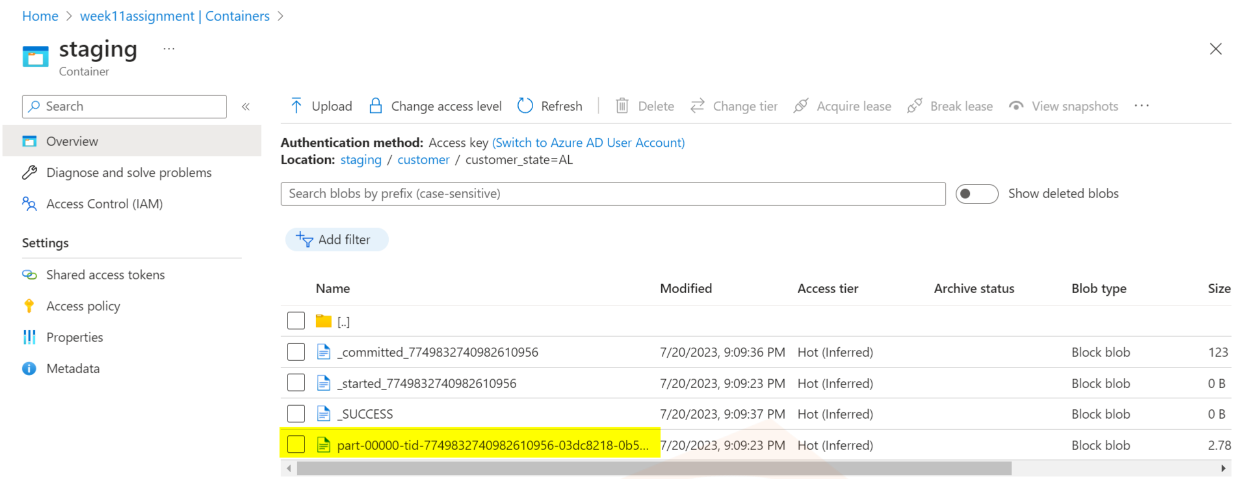
Authentication method: Access key (Switch to Azure AD User Account)
Location: staging

	Name	Modified	Access tier	Archive status	Blob type	Size
□	customer					
□	order					
□	customer	7/20/2023, 9:09:23 PM	Hot (Inferred)		Block blob	0 B
□	order	7/20/2023, 9:08:56 PM	Hot (Inferred)		Block blob	0 B

- *Staging Container Snippet - Customer folder*



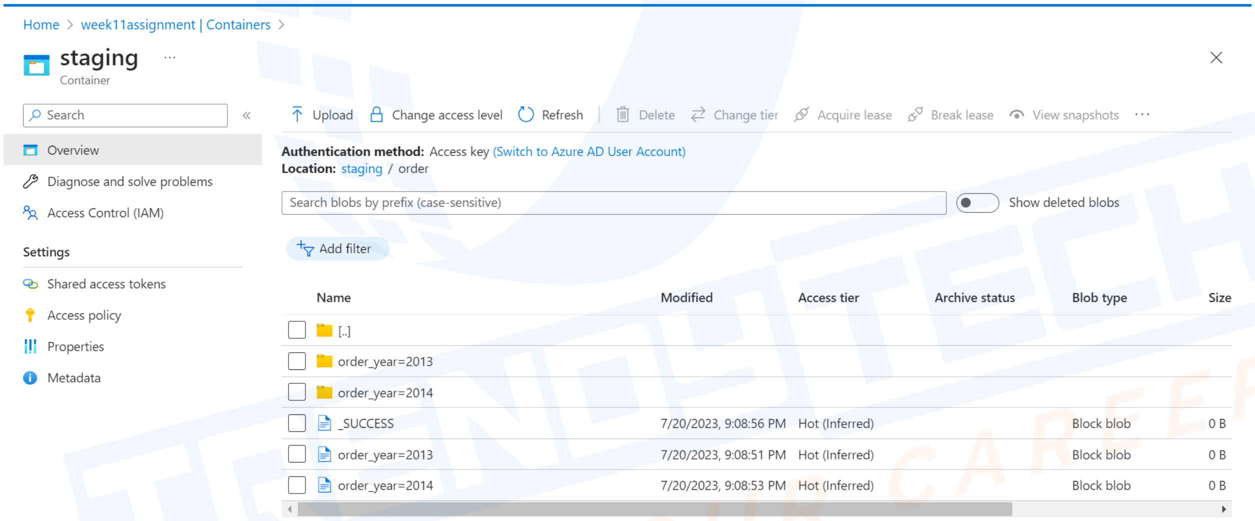
	Name	Modified	Access tier	Archive status	Blob type	Siz
□	[..]					
□	customer_state=AL					
□	customer_state=AR					
□	customer_state=AZ					
□	customer_state=CA					
□	customer_state=CO					
□	customer_state=CT					
□	customer_state=DC					
□	customer_state=DE					
□	customer_state=FL					
□	customer_state=GA					



Authentication method: Access key (Switch to Azure AD User Account)
Location: staging / customer / customer_state=AL

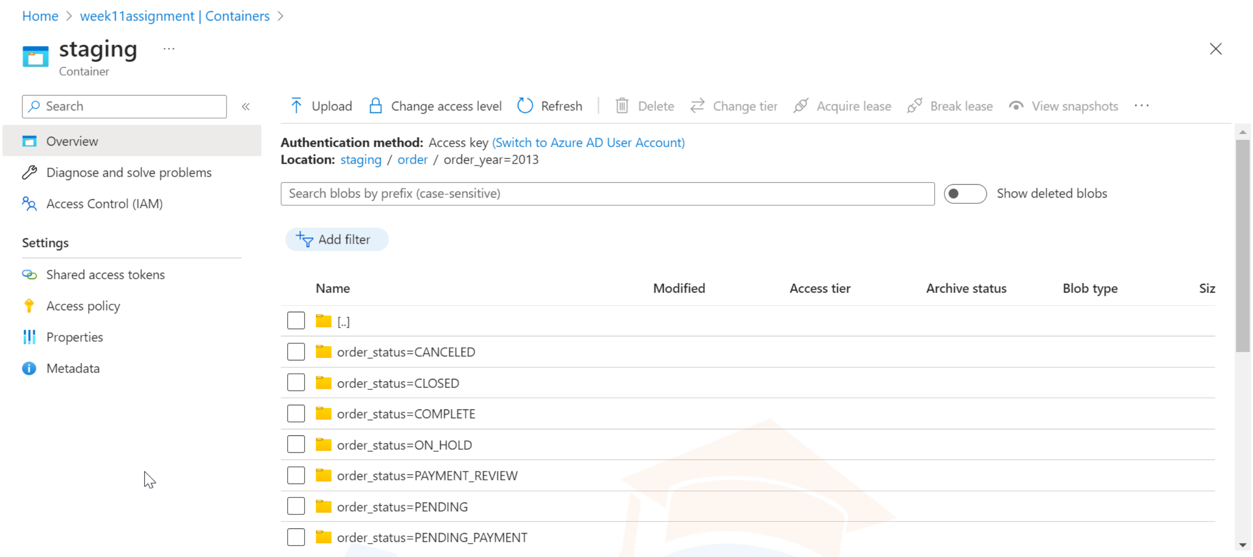
	Name	Modified	Access tier	Archive status	Blob type	Size
□	[..]					
□	_committed_7749832740982610956	7/20/2023, 9:09:36 PM	Hot (Inferred)		Block blob	123
□	_started_7749832740982610956	7/20/2023, 9:09:23 PM	Hot (Inferred)		Block blob	0 B
□	_SUCCESS	7/20/2023, 9:09:37 PM	Hot (Inferred)		Block blob	0 B
□	part-00000-tid-7749832740982610956-03dc8218-0b5…	7/20/2023, 9:09:23 PM	Hot (Inferred)		Block blob	2.78

- *Staging Container Snippet - Order folder*



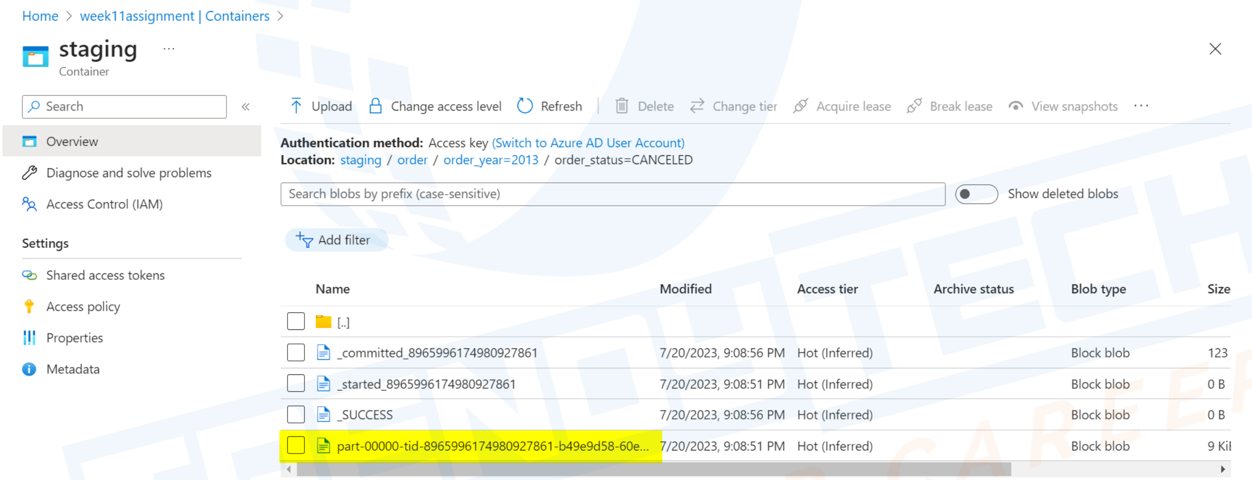
Authentication method: Access key (Switch to Azure AD User Account)
Location: staging / order

	Name	Modified	Access tier	Archive status	Blob type	Size
□	[..]					
□	order_year=2013					
□	order_year=2014					
□	_SUCCESS	7/20/2023, 9:08:56 PM	Hot (Inferred)		Block blob	0 B
□	order_year=2013	7/20/2023, 9:08:51 PM	Hot (Inferred)		Block blob	0 B
□	order_year=2014	7/20/2023, 9:08:53 PM	Hot (Inferred)		Block blob	0 B



Authentication method: Access key (Switch to Azure AD User Account)
Location: staging / order / order_year=2013

Name	Modified	Access tier	Archive status	Blob type	Siz
□ [..]					
□ order_status=CANCELED					
□ order_status=CLOSED					
□ order_status=COMPLETE					
□ order_status=ON_HOLD					
□ order_status=PAYMENT_REVIEW					
□ order_status=PENDING					
□ order_status=PENDING_PAYMENT					



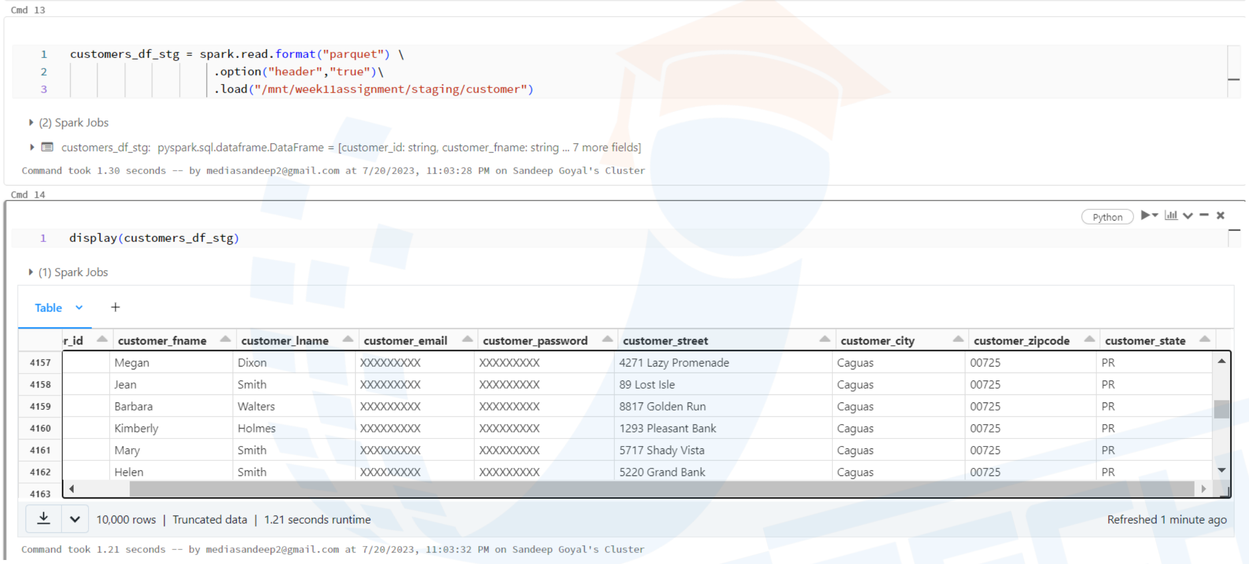
Authentication method: Access key (Switch to Azure AD User Account)
Location: staging / order / order_year=2013 / order_status=CANCELED

Name	Modified	Access tier	Archive status	Blob type	Size
□ [..]					
□ _committed_8965996174980927861	7/20/2023, 9:08:56 PM	Hot (Inferred)		Block blob	123
□ _started_8965996174980927861	7/20/2023, 9:08:51 PM	Hot (Inferred)		Block blob	0 B
□ _SUCCESS	7/20/2023, 9:08:56 PM	Hot (Inferred)		Block blob	0 B
□ part-00000-tid-8965996174980927861-b49e9d58-60e...	7/20/2023, 9:08:51 PM	Hot (Inferred)		Block blob	9 Ki

Answer 5

```
#Ans 5 - reading customer data from staging container
customers_df_stg = spark.read.format("parquet") \
                    .option("header","true")\
                    .load("/mnt/week11assignment/staging/customer")

#Ans 5 -display customers df from staging container
display(customers_df_stg)
```

```
#Ans 5 -reading orders data from staging container
orders_df_stg = spark.read.format("parquet") \
                              .option("header","true")\
                              .load("/mnt/week11assignment/staging/order")

#Ans 5 -display orders df from staging container
display(orders_df_stg)
```



Cmd 13

```
customers_df_stg = spark.read.format("parquet") \
                    .option("header","true")\
                    .load("/mnt/week11assignment/staging/customer")
```

▸ (2) Spark Jobs

▸ customers_df_stg: pyspark.sql.dataframe.DataFrame = [customer_id: string, customer_fname: string ... 7 more fields]

```
Command took 1.30 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 11:03:28 PM on Sandeep Goyal's Cluster
```

Cmd 14

```
display(customers_df_stg)
```

▸ (1) Spark Jobs

	r_id	customer_fname	customer_lname	customer_email	customer_password	customer_street	customer_city	customer_zipcode	customer_state
4157		Megan	Dixon	XXXXXXXXX	XXXXXXXXX	4271 Lazy Promenade	Caguas	00725	PR
4158		Jean	Smith	XXXXXXXXX	XXXXXXXXX	89 Lost Isle	Caguas	00725	PR
4159		Barbara	Walters	XXXXXXXXX	XXXXXXXXX	8817 Golden Run	Caguas	00725	PR
4160		Kimberly	Holmes	XXXXXXXXX	XXXXXXXXX	1293 Pleasant Bank	Caguas	00725	PR
4161		Mary	Smith	XXXXXXXXX	XXXXXXXXX	5717 Shady Vista	Caguas	00725	PR
4162		Helen	Smith	XXXXXXXXX	XXXXXXXXX	5220 Grand Bank	Caguas	00725	PR
4163									

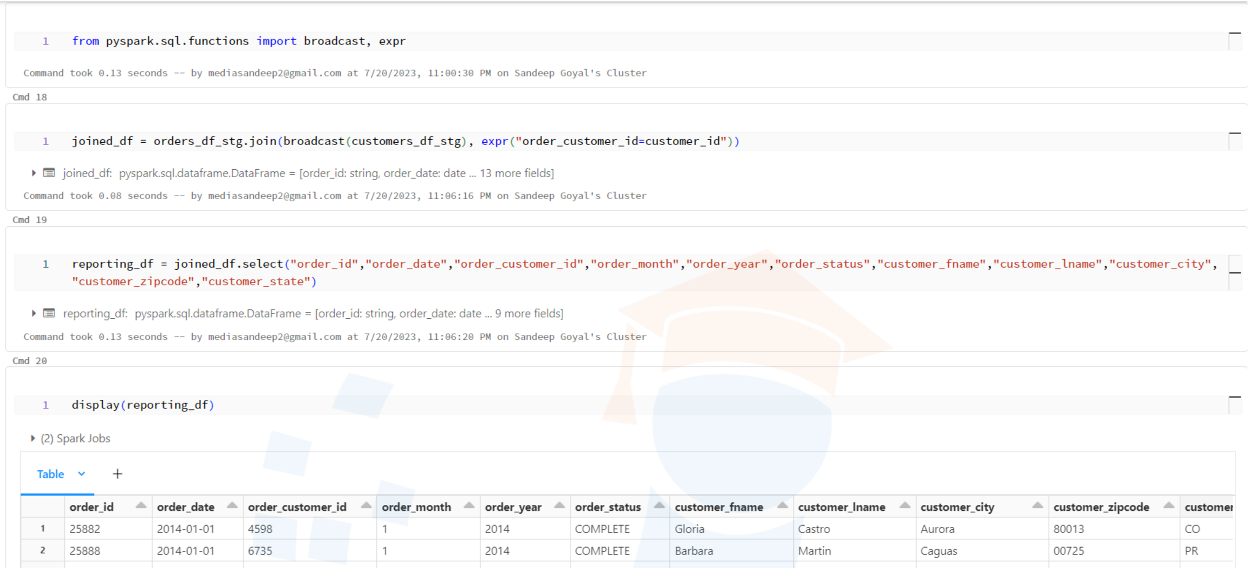
10,000 rows | Truncated data | 1.21 seconds runtime

Refreshed 1 minute ago

```
Command took 1.21 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 11:03:32 PM on Sandeep Goyal's Cluster
```

Answer 5a

```
from pyspark.sql.functions import broadcast, expr

#Ans 5a -joining orders and customers df
joined_df = orders_df_stg.join(broadcast(customers_df_stg),
expr("order_customer_id=customer_id"))

#Ans 5a - creating final reporting df
reporting_df =
joined_df.select("order_id","order_date","order_customer_id","order_month","order_year","order_status","customer_fname","customer_lname","customer_city","customer_zipcode","customer_state")

#Ans 5a - display reporting df
display(reporting_df)
```



```
from pyspark.sql.functions import broadcast, expr
```

Command took 0.13 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 11:00:30 PM on Sandeep Goyal's Cluster

Cmd 18

```
joined_df = orders_df_stg.join(broadcast(customers_df_stg), expr("order_customer_id=customer_id"))
```

joined_df: pyspark.sql.dataframe.DataFrame = [order_id: string, order_date: date ... 13 more fields]

Command took 0.08 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 11:06:16 PM on Sandeep Goyal's Cluster

Cmd 19

```
reporting_df = joined_df.select("order_id","order_date","order_customer_id","order_month","order_year","order_status","customer_fname","customer_lname","customer_city",
"customer_zipcode","customer_state")
```

reporting_df: pyspark.sql.dataframe.DataFrame = [order_id: string, order_date: date ... 9 more fields]

Command took 0.13 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 11:06:20 PM on Sandeep Goyal's Cluster

Cmd 20

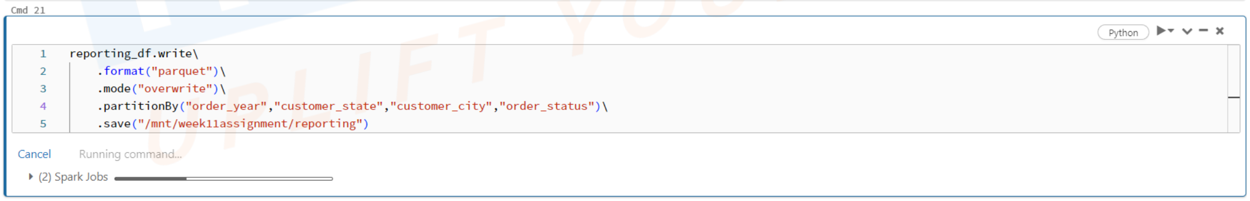
```
display(reporting_df)
```

(2) Spark Jobs

	order_id	order_date	order_customer_id	order_month	order_year	order_status	customer_fname	customer_lname	customer_city	customer_zipcode	customer
1	25882	2014-01-01	4598	1	2014	COMPLETE	Gloria	Castro	Aurora	80013	CO
2	25888	2014-01-01	6735	1	2014	COMPLETE	Barbara	Martin	Caguas	00725	PR

Answer 5b

```
#Ans 5b - writing the reporting_df into reporting container
reporting_df.write\
    .format("parquet")\
    .mode("overwrite")\
    .partitionBy("order_year","customer_state","customer_city","order_status")\
    .save("/mnt/week11assignment/reporting")
```



Cmd 21

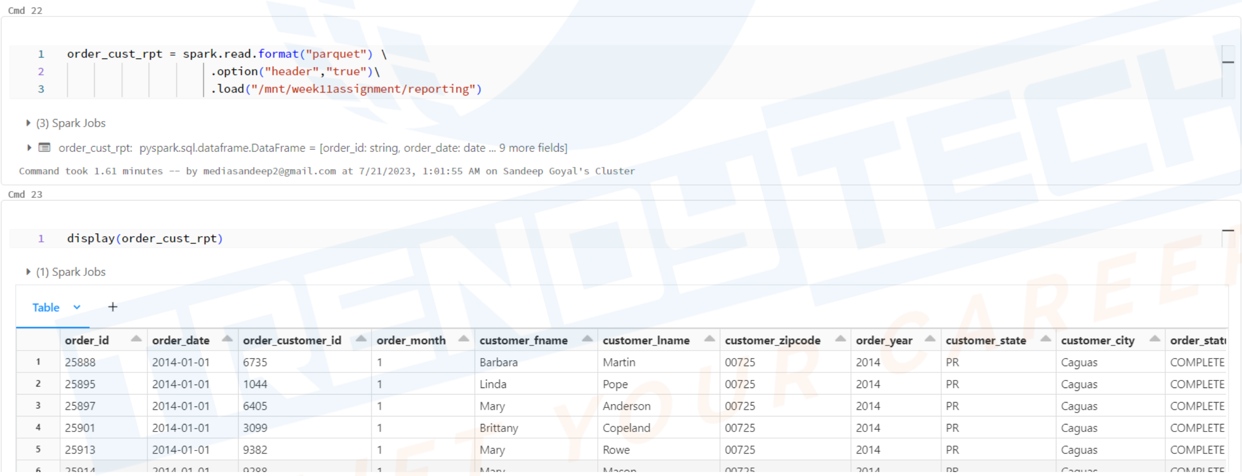
```
reporting_df.write\
    .format("parquet")\
    .mode("overwrite")\
    .partitionBy("order_year","customer_state","customer_city","order_status")\
    .save("/mnt/week11assignment/reporting")
```

Running command...

(2) Spark Jobs

Answer 6

```
#Ans 6 - reading data from reporting container to make some final views
order_cust_rpt = spark.read.format("parquet") \
                    .option("header","true")\
                    .load("/mnt/week11assignment/reporting")

#Ans 6 -  display the values in the dataframe read from reporting container
display(order_cust_rpt)
#Ans 6 - creating view from data in reporting container
order_cust_rpt.select(expr("order_customer_id").alias("customer_id"),expr("concat(customer_fname,'
',customer_lname)").alias("customer_name"),"order_date","customer_city","customer_state","order_status","order_year","order_month").createOrReplaceTempView("order_cust_vw")

#Ans 6 - doing a quick display on the view created
spark.sql("select * from order_cust_vw limit 10").show()
```



Cmd 22

```
order_cust_rpt = spark.read.format("parquet") \
                    .option("header","true")\
                    .load("/mnt/week11assignment/reporting")
```

▸ (3) Spark Jobs

▸ order_cust_rpt: pyspark.sql.dataframe.DataFrame = [order_id: string, order_date: date ... 9 more fields]

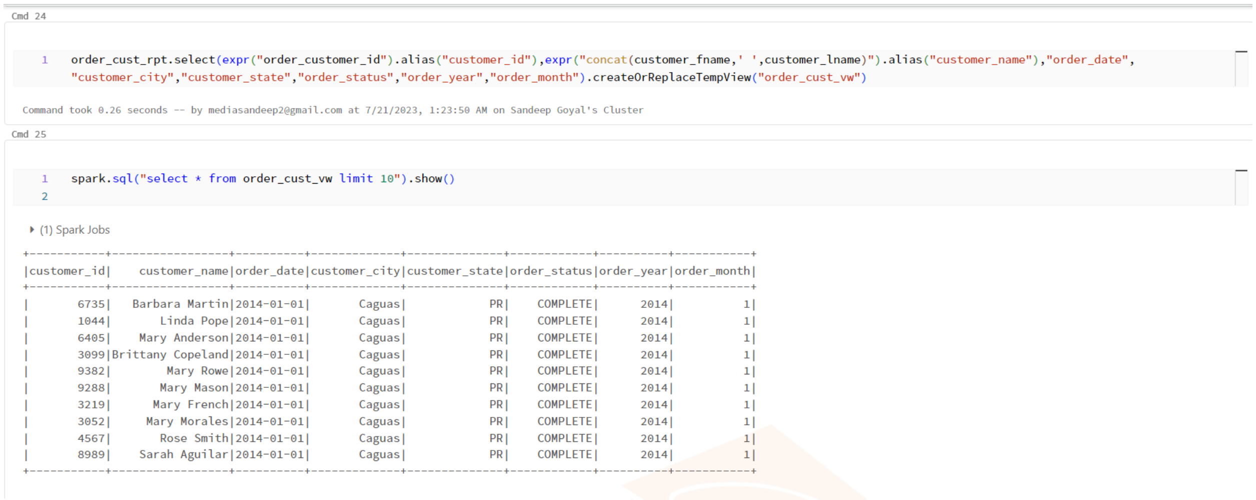
```
Command took 1.61 minutes -- by mediasandeep2@gmail.com at 7/21/2023, 1:01:55 AM on Sandeep Goyal's Cluster
```

Cmd 23

```
display(order_cust_rpt)
```

▸ (1) Spark Jobs

	order_id	order_date	order_customer_id	order_month	customer_fname	customer_lname	customer_zipcode	order_year	customer_state	customer_city	order_statu
1	25888	2014-01-01	6735	1	Barbara	Martin	00725	2014	PR	Caguas	COMPLETE
2	25895	2014-01-01	1044	1	Linda	Pope	00725	2014	PR	Caguas	COMPLETE
3	25897	2014-01-01	6405	1	Mary	Anderson	00725	2014	PR	Caguas	COMPLETE
4	25901	2014-01-01	3099	1	Brittany	Copeland	00725	2014	PR	Caguas	COMPLETE
5	25913	2014-01-01	9382	1	Mary	Rowe	00725	2014	PR	Caguas	COMPLETE
6	25914	2014-01-01	9288	1	Mary	Mason	00725	2014	PR	Caguas	COMPLETE



Cmd 24

```
order_cust_rpt.select(expr("order_customer_id").alias("customer_id"),expr("concat(customer_fname,' ',customer_lname)").alias("customer_name"),"order_date",
"customer_city","customer_state","order_status","order_year","order_month").createOrReplaceTempView("order_cust_vw")
```

Command took 0.26 seconds -- by mediasandeep2@gmail.com at 7/21/2023, 1:23:50 AM on Sandeep Goyal's Cluster

Cmd 25

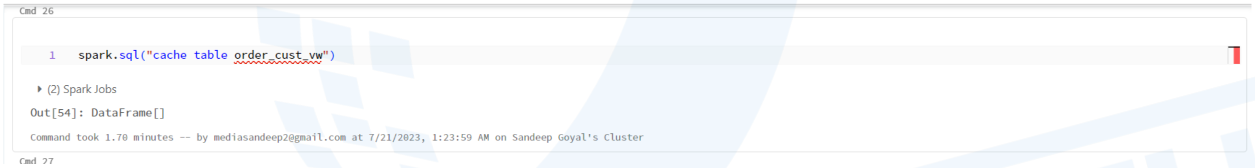
```
spark.sql("select * from order_cust_vw limit 10").show()

```

▸ (1) Spark Jobs

```
+-----------+---------------+----------+-------------+--------------+------------+----------+-----------+
|customer_id|  customer_name|order_date|customer_city|customer_state|order_status|order_year|order_month|
+-----------+---------------+----------+-------------+--------------+------------+----------+-----------+
|       6735| Barbara Martin|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       1044|     Linda Pope|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       6405|  Mary Anderson|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       3099|Brittany Copeland|2014-01-01|     Caguas|            PR|    COMPLETE|      2014|          1|
|       9382|      Mary Rowe|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       9288|     Mary Mason|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       3219|    Mary French|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       3052|   Mary Morales|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       4567|     Rose Smith|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
|       8989|  Sarah Aguilar|2014-01-01|       Caguas|            PR|    COMPLETE|      2014|          1|
+-----------+---------------+----------+-------------+--------------+------------+----------+-----------+
```

Answer 6a

```
#Ans 6a - cache the view created
spark.sql("cache table order_cust_vw")
```



Cmd 26

```
spark.sql("cache table order_cust_vw")
```

▸ (2) Spark Jobs

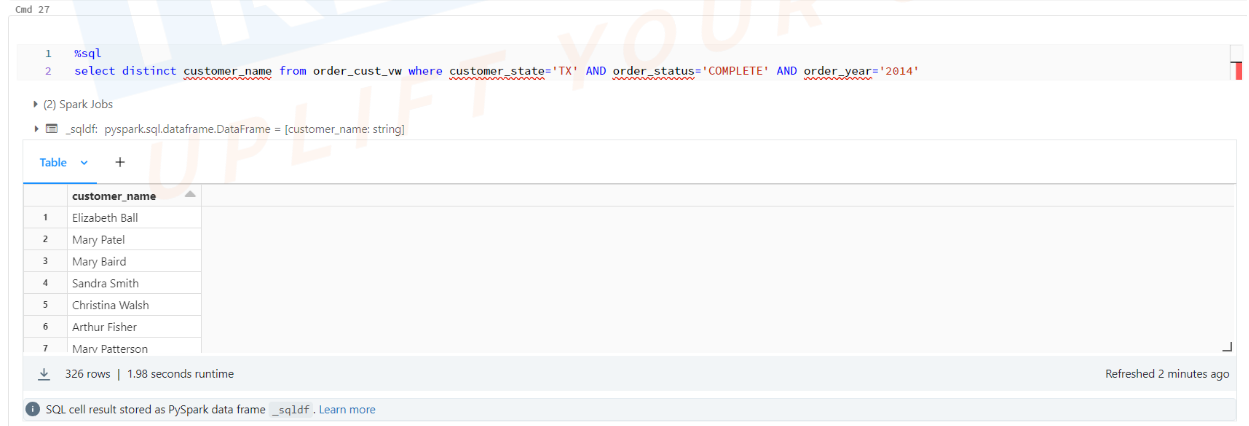
Out[54]: DataFrame[]

Command took 1.70 minutes -- by mediasandeep2@gmail.com at 7/21/2023, 1:23:59 AM on Sandeep Goyal's Cluster

Cmd 27

Answer 6b

```
%sql
--Ans 6b - running the required query
select distinct customer_name from order_cust_vw where customer_state='TX'
AND order_status='COMPLETE' AND order_year='2014'
```



Cmd 27

```
%sql
select distinct customer_name from order_cust_vw where customer_state='TX' AND order_status='COMPLETE' AND order_year='2014'
```

▸ (2) Spark Jobs

▸ _sqldf: pyspark.sql.dataframe.DataFrame = [customer_name: string]

	customer_name
1	Elizabeth Ball
2	Mary Patel
3	Mary Baird
4	Sandra Smith
5	Christina Walsh
6	Arthur Fisher
7	Marv Patterson

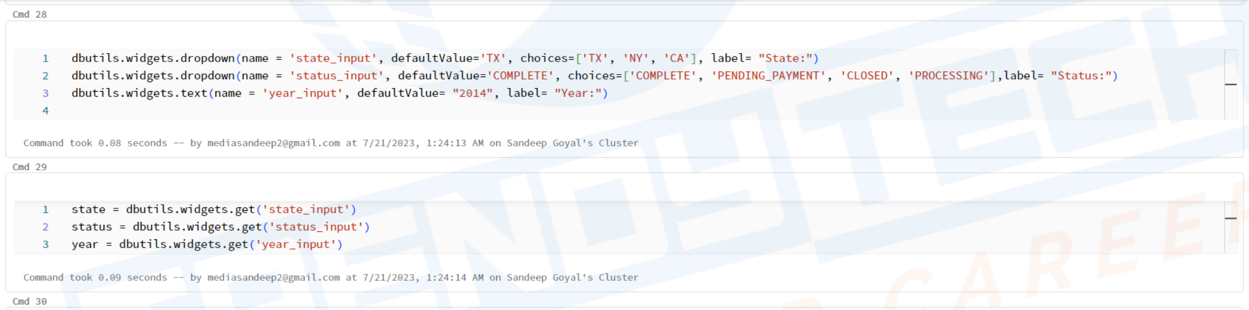
326 rows | 1.98 seconds runtime

Refreshed 2 minutes ago

SQL cell result stored as PySpark data frame _sqldf . Learn more

Answer 6c

```
#Ans 6c - creating state, status and year widget
dbutils.widgets.dropdown(name = 'state_input', defaultValue='TX',
choices=['TX', 'NY', 'CA'], label= "State:")
dbutils.widgets.dropdown(name = 'status_input', defaultValue='COMPLETE',
choices=['COMPLETE', 'PENDING_PAYMENT', 'CLOSED',
'PROCESSING'],label= "Status:")
dbutils.widgets.text(name = 'year_input', defaultValue= "2014", label= "Year:")


#Ans 6c - get the values from widgets into variables
state = dbutils.widgets.get('state_input')
status = dbutils.widgets.get('status_input')
year = dbutils.widgets.get('year_input')

#Ans 6c - display the results of the query with values coming from widgets
display(spark.sql(f"SELECT distinct customer_name FROM order_cust_vw
WHERE customer_state = '{state}' AND order_status = '{status}' AND
order_year = {year}"))
```



Cmd 28

```
dbutils.widgets.dropdown(name = 'state_input', defaultValue='TX', choices=['TX', 'NY', 'CA'], label= "State:")
dbutils.widgets.dropdown(name = 'status_input', defaultValue='COMPLETE', choices=['COMPLETE', 'PENDING_PAYMENT', 'CLOSED', 'PROCESSING'],label= "Status:")
dbutils.widgets.text(name = 'year_input', defaultValue= "2014", label= "Year:")

```

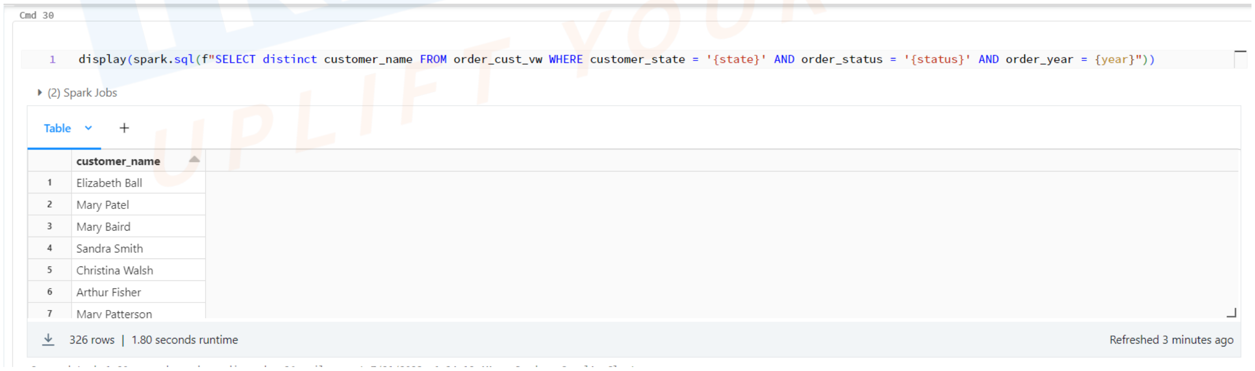
Command took 0.08 seconds -- by mediasandeep2@gmail.com at 7/21/2023, 1:24:13 AM on Sandeep Goyal's Cluster

Cmd 29

```
state = dbutils.widgets.get('state_input')
status = dbutils.widgets.get('status_input')
year = dbutils.widgets.get('year_input')
```

Command took 0.09 seconds -- by mediasandeep2@gmail.com at 7/21/2023, 1:24:14 AM on Sandeep Goyal's Cluster

Cmd 30



Cmd 30

```
display(spark.sql(f"SELECT distinct customer_name FROM order_cust_vw WHERE customer_state = '{state}' AND order_status = '{status}' AND order_year = {year}"))
```

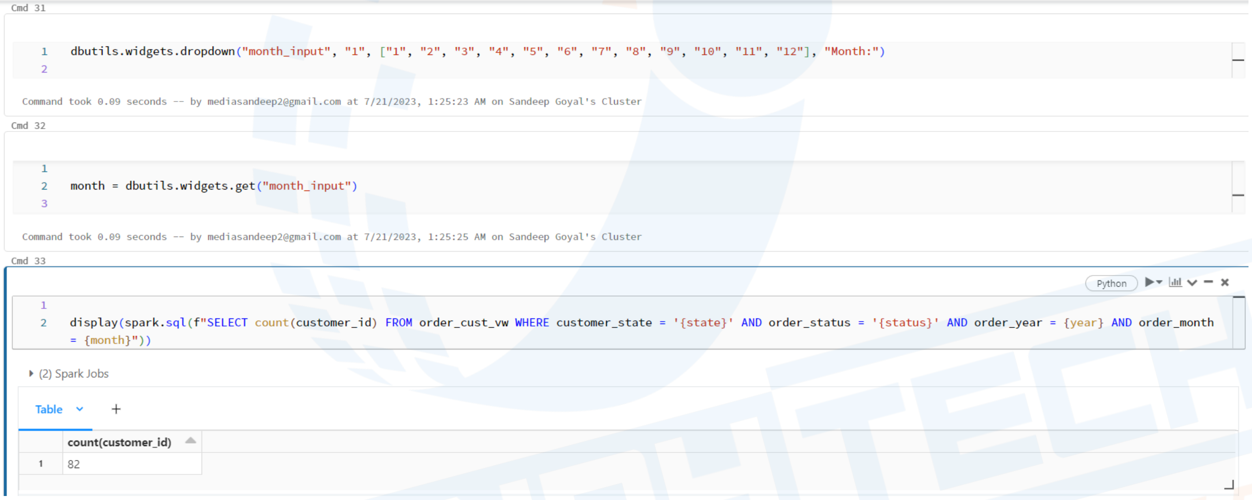
	customer_name
1	Elizabeth Ball
2	Mary Patel
3	Mary Baird
4	Sandra Smith
5	Christina Walsh
6	Arthur Fisher
7	Mary Patterson

326 rows | 1.80 seconds runtime

Refreshed 3 minutes ago

Answer 6d

```
#Ans 6d - creating month widget
dbutils.widgets.dropdown("month_input", "1", ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"], "Month:")


#Ans 6d - get the values from month widget
month = dbutils.widgets.get("month_input")

#Ans 6d - run a query with month from widget
display(spark.sql(f"SELECT count(customer_id) FROM order_cust_vw WHERE customer_state = '{state}' AND order_status = '{status}' AND order_year = {year} AND order_month = {month}"))
```



Cmd 31

```
dbutils.widgets.dropdown("month_input", "1", ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"], "Month:")

```

Command took 0.09 seconds -- by mediasandeep2@gmail.com at 7/21/2023, 1:25:23 AM on Sandeep Goyal's Cluster

Cmd 32

```

month = dbutils.widgets.get("month_input")

```

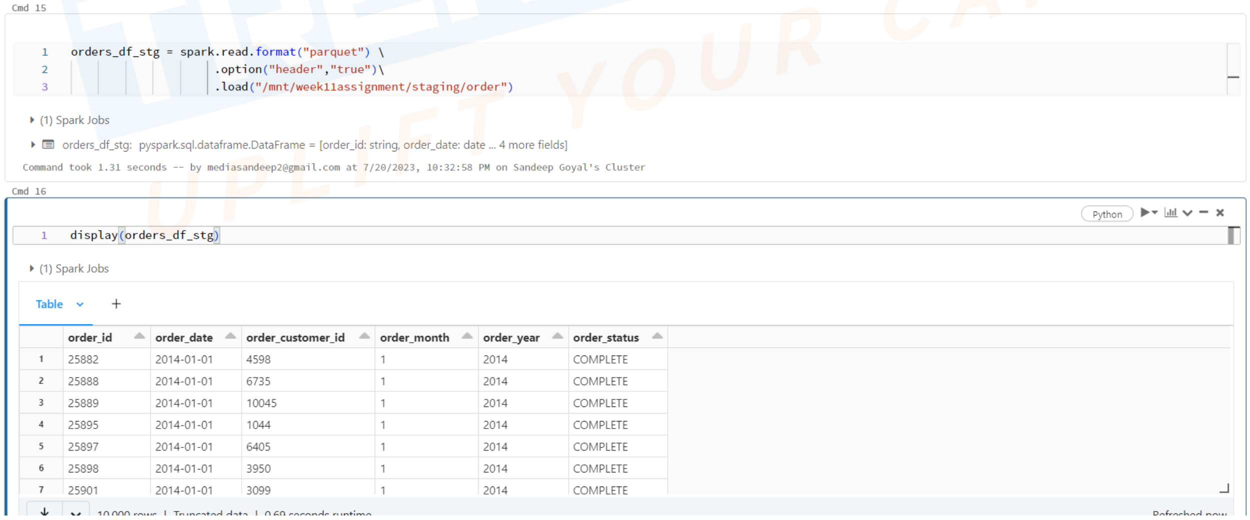
Command took 0.09 seconds -- by mediasandeep2@gmail.com at 7/21/2023, 1:25:25 AM on Sandeep Goyal's Cluster

Cmd 33

```

display(spark.sql(f"SELECT count(customer_id) FROM order_cust_vw WHERE customer_state = '{state}' AND order_status = '{status}' AND order_year = {year} AND order_month = {month}"))
```

(2) Spark Jobs

	count(customer_id)
1	82



Cmd 15

```
orders_df_stg = spark.read.format("parquet") \
                          .option("header","true")\
                          .load("/mnt/week11assignment/staging/order")
```

(1) Spark Jobs

orders_df_stg: pyspark.sql.dataframe.DataFrame = [order_id: string, order_date: date ... 4 more fields]

Command took 1.31 seconds -- by mediasandeep2@gmail.com at 7/20/2023, 10:32:58 PM on Sandeep Goyal's Cluster

Cmd 16

```
display(orders_df_stg)
```

(1) Spark Jobs

	order_id	order_date	order_customer_id	order_month	order_year	order_status
1	25882	2014-01-01	4598	1	2014	COMPLETE
2	25888	2014-01-01	6735	1	2014	COMPLETE
3	25889	2014-01-01	10045	1	2014	COMPLETE
4	25895	2014-01-01	1044	1	2014	COMPLETE
5	25897	2014-01-01	6405	1	2014	COMPLETE
6	25898	2014-01-01	3950	1	2014	COMPLETE
7	25901	2014-01-01	3099	1	2014	COMPLETE

10,000 rows | Truncated data | 0.69 seconds runtime

Refreshed now