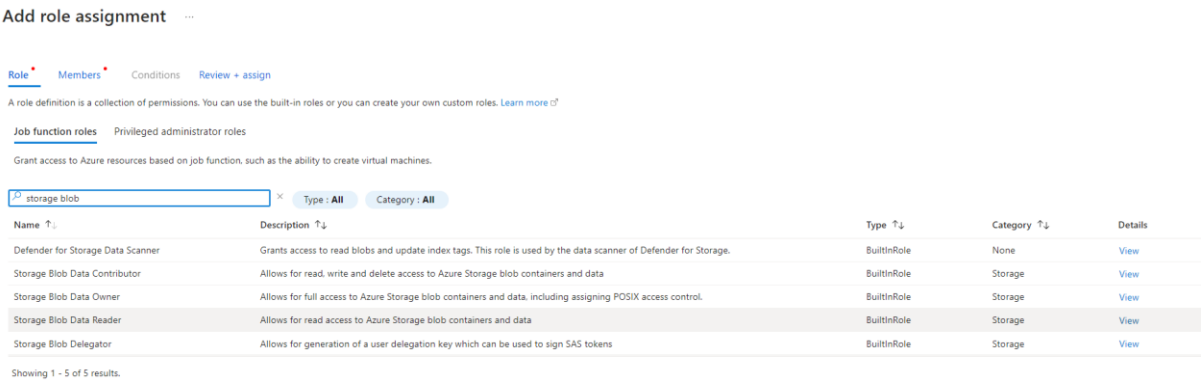
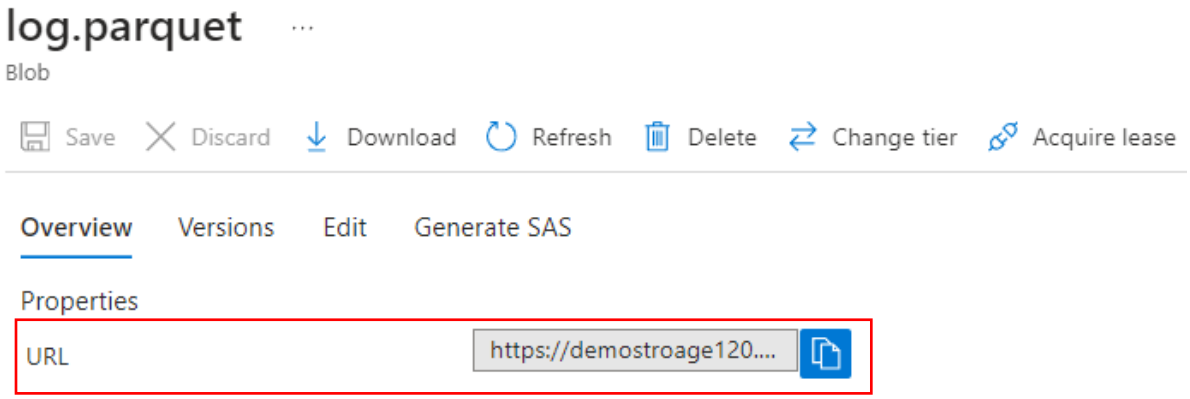
😊 Spark Pool Load Data

In this lab, we're configuring our Spark pool to load data from Azure Data Lake Gen 2 storage into our notebook. The end goal is to access and analyze data stored in our storage account using Spark within Azure Synapse Analytics. We set permissions, retrieve the storage container's URL, and then write Python code to load the data into our Spark pool. Additionally, we perform basic data manipulation tasks like selecting specific columns, filtering out null values, and counting rows with null values to gain insights into the data's quality and structure. Ultimately, this lab facilitates seamless data access and analysis workflows within Azure Synapse Analytics.

1. Now in this lab let's write a notebook that can be used to read data from our Azure Data Lake Gen 2 storage account.
2. But first you need to go to your storage account open IAM and click on add role assignment then search for storage blob data reader.
3. Then you need to assign this role to your storage account so that Spark pool can read the data stored in your container.



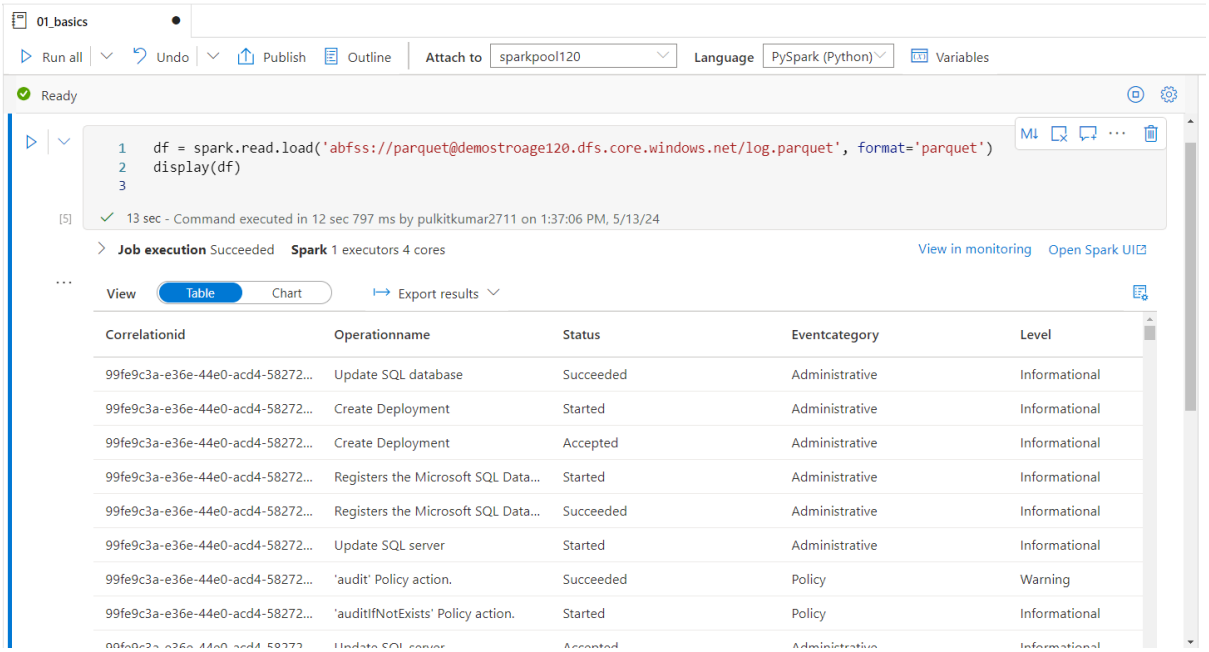
4. Once you are done assigning the role then you need to go the parquet container and copy the URL of it.
5. Copy it in your Notepad. Then move to Spark Pool.



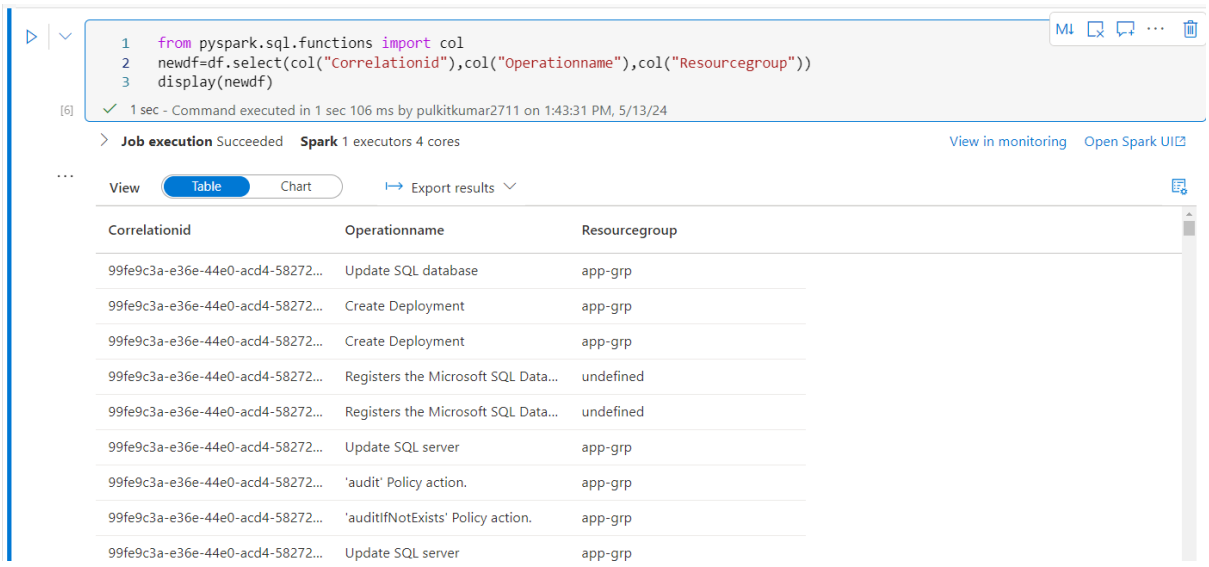
6. Now in your spark pool you need to write the code shown below here you need to change the storage account name, which is highlighted, and you need to check with the highlighted file name is correct or not.

```
df =
spark.read.load('abfss://parquet@demostroage120.dfs.core.windows.net/log.parquet', format='parquet')
display(df)
```

7. Now you need to run this statement of code and you will be able to see your data from your storage account directly from your spark pool.



8. Now let's say you need to see on the selected columns then you can run the code below and you will get the data.



9. Also, you can run the code to filter out the values that should you null values. And below you can see that we go the Undefined values.

```
from pyspark.sql.functions import col
nulldf=df.filter(col("ResourceGroup").isNull())
display(nulldf)
```

✓ 1 sec - Command executed in 1 sec 87 ms by pulkitkumar2711 on 1:45:05 PM, 5/13/24

> **Job execution** Succeeded **Spark** 1 executors 4 cores

Resourcetype	Resourcegroup	Resource
undefined	undefined	/subscriptions/6912d7a0-bc28-4...
undefined	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...
Microsoft.RecoveryServices/locat...	undefined	/subscriptions/6912d7a0-bc28-4...

10. Also, we can run this code to check how many rows are undefined or null and we got the output.

```
rows=nulldf.count()
print(f"The number of rows : {rows}")
```

[8] ✓ 1 sec - Command executed in 1 sec 820 ms by pulkitkumar2711 on 1:47:28 PM, 5/13/24

> **Job execution** Succeeded **Spark** 1 executors 4 cores

```
The number of rows : 191
```

11. Now you can also use group by clause to summarize your data.

```
1  summaryRows=df.groupBy("ResourceGroup").count()
2  display(summaryRows)
```

[9] ✓ 3 sec - Command executed in 2 sec 782 ms by pulkitkumar2711 on 1:49:03 PM, 5/13/24

Job execution Succeeded **Spark** 1 executors 4 cores View in monitoring Open Spark UI

View: Table | Chart — Export results

ResourceGroup	count
app-grp-asr	200
DESTINATION-GRP	3
log-grp	62
site-recovery-vault-rg	82
DefaultResourceGroup-EUS	21
MC_app-grp_appcluster_northeu...	93
mc_app-grp_appcluster_northeur...	12
undefined	191
MC_APP-GRP_APPCLUSTER_NOR...	4