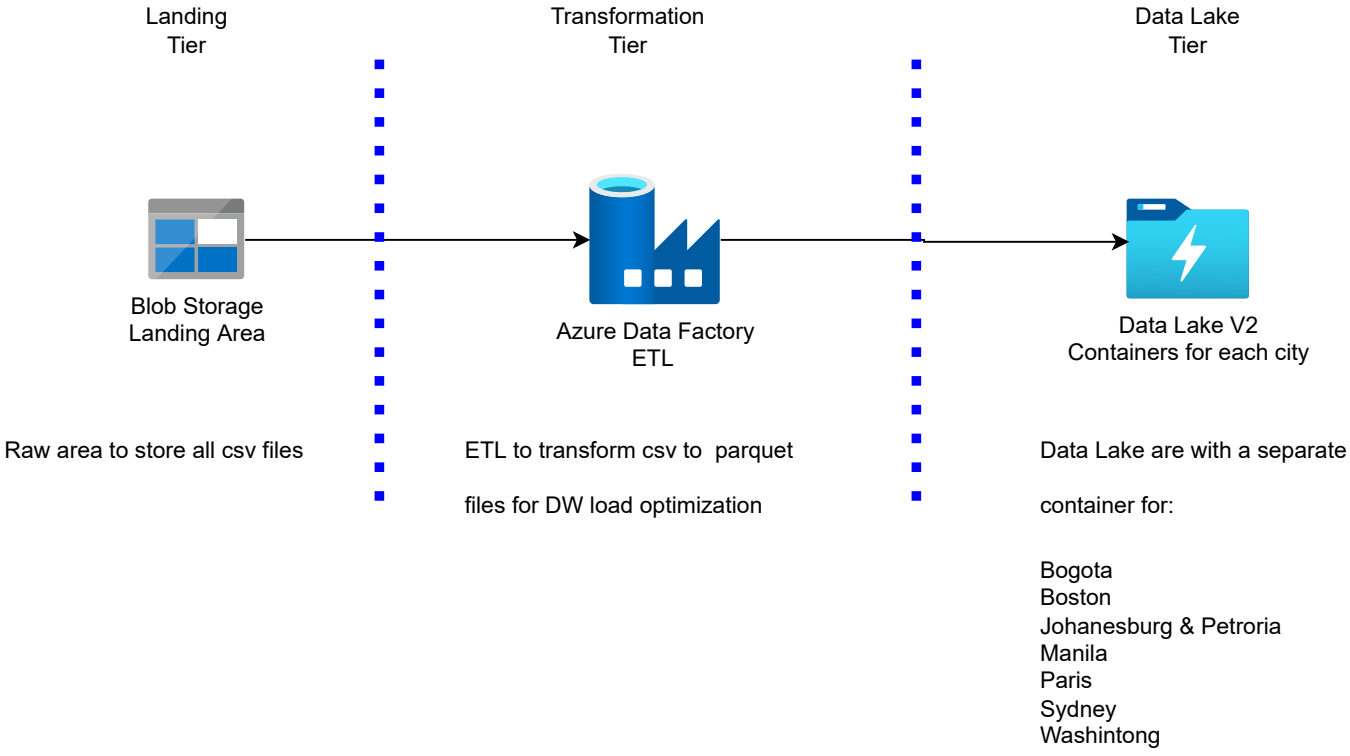
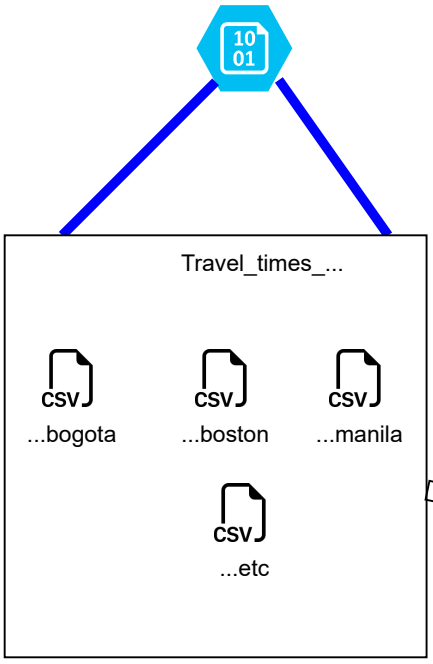
Travel Time Uber Movement
General Architecture



We tackle the problem with the creation of 3 tiers 2 for storage and 1 for processing. The first tier is the landing tier where is going to be our staging storage to lad the CSV files to be consumed by ETL process that are going to transform them into .parquet files for better performance in the querying and ingestion into the DWH. The last tier is the datalake where we are going to use the hierarchical level to store the newly created parquet files

Blob Storage Landing Zone



Security

Secure Transfer (Https)

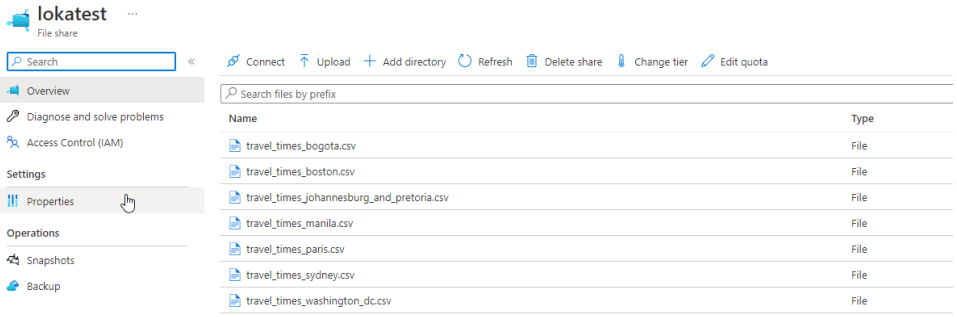
Public access: disabled

Azure Active Directory Authorization

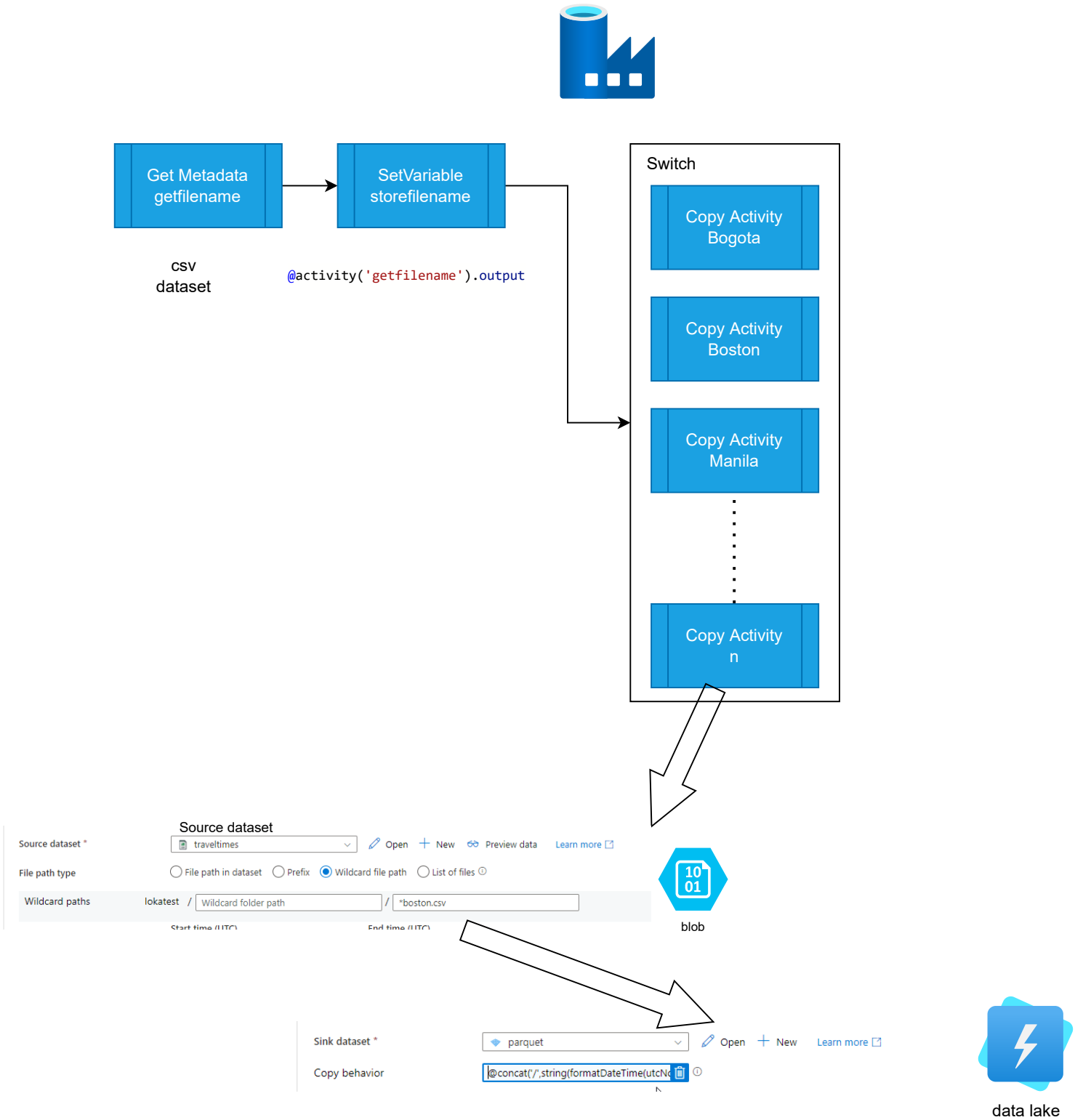
TLS V1.2

Encryption Microsoft-Managed keys MMK

Encryption Infrastructure enabled



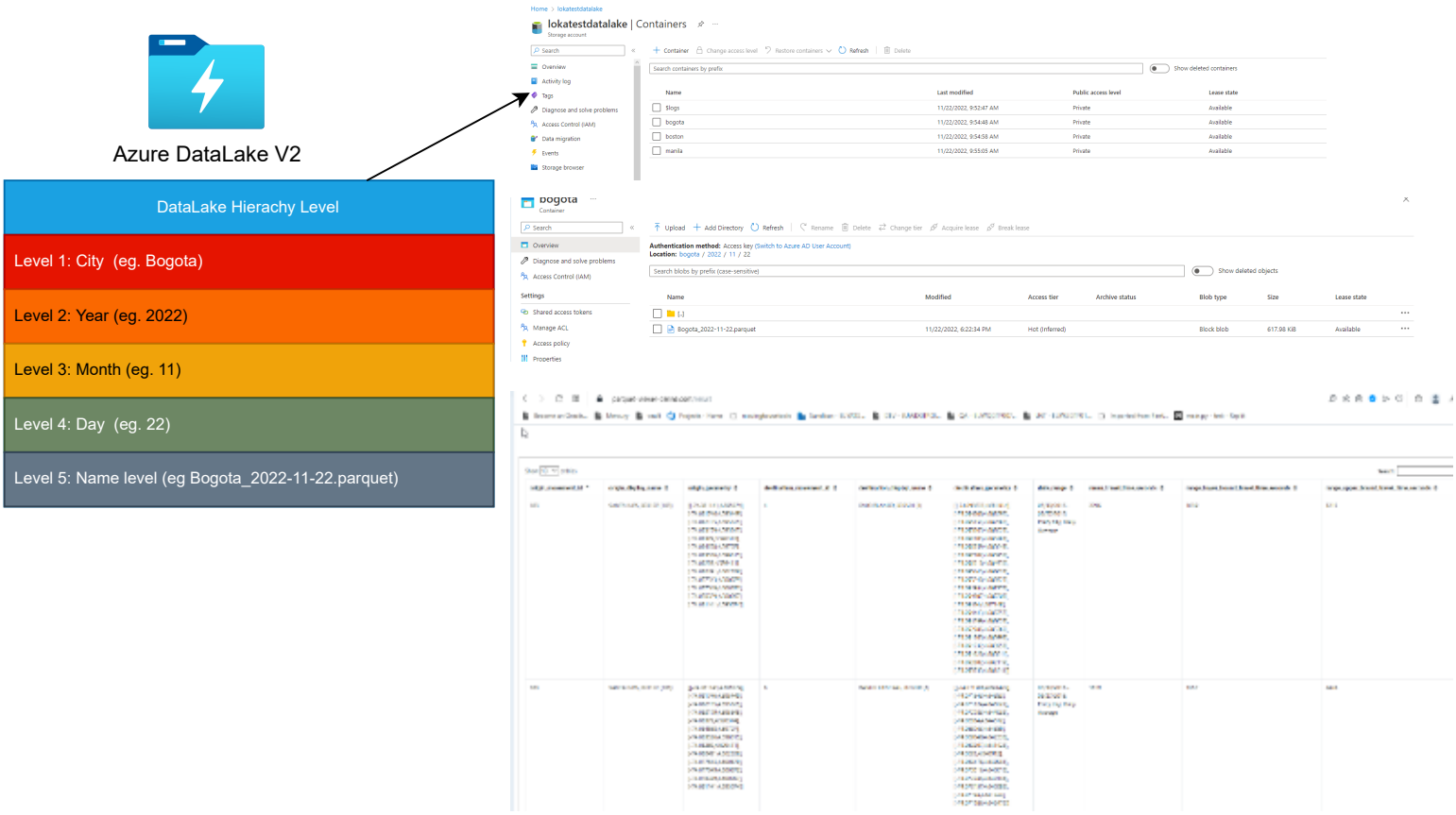
The purpose of the landing zone is to leverage the basic blob storage service provided by Azure in order to stage the raw csv file with the uber movements data. No hierarchy is needed here as per this is going to be populated by a simple transfer service in order to be consumed by ADF (see previous page)




```
@concat('/',string(formatDateTime(utcNow(),'yyyy')),'/',string(formatDateTime(utcNow(),'yyyy')),'/',string(formatDateTime(utcNow(),'MM')),
                '/',string(formatDateTime(utcNow(),'dd')),'/Bogota_', formatDateTime(utcNow(), 'yyyy-dd-MM'),'. parquet ')
```

ETL Process

Leveraging Azure Data Factory 2 data sets are required. 1 for the source (blob shares) and a second for the destination (data lake). ETL is going to read the file, get the name, validate the last portion of the string to check what city belongs to, with this name it is going to process the copy activity assigning a hierarcy path using the city name + date + a specific file name. In this process ADF converts the file to parquet using a data set for this purpose



Security

Private enpoints, vpn, and IP addresses

MS Network Routing

Infrastructure Encryption Enabled

Encryption type : Microsoft - managed keys

DataLake features

Files are transformed from source shares from CSV to PARQUET. The reason for this is to leverage the advantage of the columnar stored format for speeding analytics queries or for data ingestion to the DWH (in this exercise we are assuming we are going to ingest to AZ synapse analytics that as well is a columnar storage). We can do query directly via any BI tool for example Power BI with the parquet connector.