# Part 1 – Environment setup

## Setup Azure

Create RG

Create ADF

Create Storage Account, make bronze container

Create Databricks

Automatically makes its own rg for underlying databricks resources

Create Synapse

Account name is the storage account name, if it doesn’t come up via subscription then manually via URL and put storage account name in

[FYI: When setting up a workspace in Azure Synapse and selecting a Data Lake Storage Gen2 file system, the **"File system name"** is a critical component. It essentially acts as a container within your storage account where data, logs, and job outputs are stored.]

File system name just make it something relevant and meaningful

Create KV

## Setup SQL On-Prem

Download sql server

download ssms

(if no work, sql server config mgr and then run the service)

download adventureworks

move to C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\Backup

restore db

[[ config mgr if you shut down ]]

create login sql script to get username and password

execute in correct db (might need to load again)

give user permissions via role on LHS

create key vault -> create secret username, password

(issue: The operation is not allowed by RBAC. If role assignments were recently changed, please wait several minutes for role assignments to become effective.

solution:

step1: select the Resource group where creating Azure Key Vault -> select "Access Control(IAM) ->Add "Add role assignment" and for Role search for "Key Vault Administrator" -> select the member by searching name or email.

step2: back to same Resource group -> "Access Control(IAM).

step3: select "view my access" you will find role created.

step4: try creating Azure secret done.

## Setup PowerBI

download in Microsoft store.

If you don’t have windows, you could try vm but nightmare – just use powerbi in synapse.

# Data Ingestion w ADF (p1)

Launch Data Factory

Install self host integration runtime to our machine (since we are running the sql server)

Go to manager -> integration runtimes

One already exists to let cloud resources integrate

New -> azure -> self hosted -> create

Manual downloads an app with key used later to run, instead do express

(if it fails do manual…)

Open integration runtime config mgr to confirm

Step 1 – connect to on prem db and copy using data factory

Create new pipeline in author

New copy data activity

Create new source dataset -> sql -> linked service (needed to connect to any data source)

Linkedservice: name, runtime, server name and db name (from ssms),

sql auth (password from kv – linkedservice, test connection)

fails cos of authentication to read

got to kv -> iam -> role assignment -> key vault secret user -> member (used manage services)

go back and select password, test connection, then create

won’t work cos you need to right click on your server in ssms and properties and security and change server authentication to SQL

restart server in config mgr then go back test connection and create (oh make sure encrypt is optional!! Or get https cert error)

then create new sink dataset, new linkedservice, your storage account may get error cos of soft delete – so go to storage account -> data protection -> uncheck enable soft delete for blobs

if this doesn’t work you can check by previewing and then run the following  
  
USE AdventureWorksLT2022;

GRANT SELECT ON SalesLT.Address TO milan;

If still doesn’t work (JreNotFound) it may be that you need java installed (via choco or brew ideally)

# Data Ingestion w ADF (p2)

Delete the file, as now creating pipeline for all tables

Create new pipeline

Create new SQL script in SSMS that lists all tables under SalesLT schema

SELECT

s.name AS SchemaName,

t.name AS TableName

FROM sys.tables t

INNER JOIN sys.schemas s

ON t.schema\_id = s.schema\_id

WHERE s.name = 'SalesLT'

So on pipeline create lookup activity, settings make a new source dataset and don’t select a specific table and use query option and copy the script (and uncheck first row only)

Run debug and look at inputs outputs on output – see its in json

Create forecah activity and connect on success

On settings click items -> dynamic -> activity outputs for look for all tables -> add .values (which is the json list output)

Update activities -> click pencil -> in foreach place copydata -> use SqlDBTables but select query and add dynamic content and insert:

@{concat('SELECT \* FROM ', item().SchemaName, '.', item().TableName )} // remember the space after from!!

Sink select the same parquet

We want it in format bronze/Schema/Tablename/Tablename.parquet so we make a new Parquet sink and select parameters where we can leverage the item() we used for the source. Now go back to the sink and update value to dynamic content and put in the relevant item() – make sure to use @

Now go back to parquet and under connection -> file path, update directory to @{concat( <<schema>>, ‘/’, <<table>>)}

And for file concat the tablename and .parquet

Validate and publish, go back to outer pipeline

We can either debug or trigger, so lets add trigger to trigger now

Click on link and can go monitor pipeline, each foreach is running concurrently as seen on gantt (if you need to make any changes, ensure you publish before triggering)

NEED TO UPDATE SSMS QUERY FOR milan PRIVILEGES:

USE AdventureWorksLT2022;

GRANT SELECT ON SCHEMA::SalesLT TO milan;

Since we made a change not in azure, we can click rerun pipeline in top left

Now you can see the files and directories in storage account

* If you get an empty file:  
  “Azure blob storage does not support having empty folders. Thus, when you try to create folders (or empty folders), there will be a duplicate empty file.
* It is a blob storage with hierrachial namespace disabled-is that the cause? Yes, enabling hierarchical workspace will enable azure data lake which supports file and directory semantics and therefore which wouldn't create that additional file.”
* E.g. <https://stackoverflow.com/questions/76074718/additional-empty-blob-created-with-folder-names-in-azure-storage-container-not-a>

# Data Transformation w Databricks (p1 – set it up and mount the data lake)

Rg -> databricks -> workspace

Workspace is a place for notebooks

Compute is for spinning up clusters to run your notebooks on (since you could be using vast amounts of data)

Workflows is used to create jobs of notebooks from workspace, we wont use as will use DF pipelines to trigger instead

First spin up compute / spark cluster

So new compute, change name, single node, single user, change daabricks runtime version to default standard, node type to smallest, keep terminate checked and change to 10/15 min (only problem is debugging)

Expand advanced – enable credential pass through

WHY? Go to storage account -> IAM -> role assignments -> storage blob data contributors

These are based on managed identities which are default id’s assigned to resources, cant do a databricks one though so add yourself (its not automatic)

2 solutions: mount – which you can see below, or enabling this via email account up top right

anyway create and then whilst it’s building (could take 5), lets make a notebook in workspace -> shared -> create -> notebook

name storagemount, choose python, select cluster

to mount using credential passthrough go to: <https://learn.microsoft.com/en-us/azure/databricks/archive/credential-passthrough/adls-passthrough>

and copy in python code

configs = {

"fs.azure.account.auth.type": "CustomAccessToken",

"fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")

}

# Optionally, you can add <directory-name> to the source URI of your mount point.

dbutils.fs.mount(

source = "abfss://<container-name>@<storage-account-name>.dfs.core.windows.net/",

mount\_point = "/mnt/<mount-name>",

extra\_configs = configs)

then change container to bronze, sa to the name, mount to bronze and run

now we can access the data via mount point

theres a databricks file system fuction dbutils.fs.ls("/mnt/bronze") we can use to make sure we’re working

do the same for silver and gold, lets make those folders and use the same databricks code (don’t have to but will make it easier for us to work with

# Data Transformation w Databricks (p2)

## Bronze to silver (L1 transformation)

Let’s check the data in ssms to see how it’s looking, we can do this by right clicking on a table and select top 1000 rows then execute

Looks clean and is structured

Lets change ModifiedDate to just a date In ProductDescr., then in ProductId there are a bunch of dates, so lets change for all that have datetime to just date

Go to databricks and create 2 new notebooks, bronze to silver (for this section) and silver to gold (for next)

In bronze to silver, paste in the code line by line

df is temp view with table structure with all schema specific to the source file

can use various languages in the same notebook (magic command) e.g. markdown, and e.g. SQL

the first piece of code changes modifieddate to date type, the next chunk changes all date cols to that type using a foreach loop

save in delt – built on parquet – developed b databricks, so all parquet features + more – can track version history and schema changes

easily handles change in schema

open silver and see that running the code puts new dataframes there

## Silver to gold (L2 transformation)

Generally joining / aggregation to get final facts and dimensions, and col name changes

Les changecol names from pascal case to upper snake case

The replace code makes a new column name and adds in the under score before the n+1 capital, then replaces the original column name

Modularise this such that it can be used as a function for all

Now when the foreach runs, you’ll see spark jobs running in parallel, the one and only thing that spark runs in parallel are the tasks for each stage

For debugging ensure that on the blue square with play button you select the drop down to the run below, to run all cells from this point

# Data Transformation w Databricks (p3 – ADF notebook pipeline)

Go to data factory studio, go to copy\_all\_tables

Need to update to include data bricks notebook

First establish connection between them via linked service (under manage tab), add new search databricks under compute

Autointegration runtime as it is an azure service

For cluster: specify existing, lets do access token and we will use azure key vault since it’s sensitive data

Go to databricks, click username -> settings -> developer -> new token -> generate -> Copy token

Go to key vault -> your key vault -> secret -> new -> create name and paste password -> create

Go back to studio, select the key vault, and the new name, choose the existing cluster we made, test connection, create, publish all to save

Now go back to factory resources, select our pipeline, create a notebook activity (we will do 2, one for each), but lets do one first

Rename bronze to silver, add on successful from the foreach

Now click on the activity, select the linked service under azure databricks, then browse the notebook path and select the bronze to silver one

Do the same for the silver to gold and have activity as from successful

We don’t need to do storage mount as we’ve ran it manually it doesn’t need to be run again

Publish all, test pipeline using trigger now

Our databricks notebooks have mode as overwrite but could have incremental updates, we will cover in a future video

When we get to the databricks portion, hover over the activity name in the bottom pane and you’ll see glasses icon, this opens details. Open these details and open run page url. This lets us see the notebook running in real time (letting us monitor / debug in real time) so sometimes the display functions can be kept in the notebook so you can see what happening

Whilst gold is running if we go to the silver we can see that it’s processed twice, but the thing with delta format is it has a version tracking feature so you get the most recent total version of the data

# Data Loading using Synapse Analytics

Go to the synapse, as you can see it looks like data factory, that’s because its built on top of it – meaning it can do pretty much what data factory can do (e.g. pipelines and monitoring), but additionally it has data and develop tabs

Data can be used to create databases (not available in ADF)

In develop you can create the sql scripts / notebooks (similar to databricks)

Lets create the db, go to data -> + -> SQL DB -> serverless sql pool -> gold\_db name -> create

The concept of a "pool" in this context is related to the idea of pooling resources to provide a scalable, high-performance environment for data processing and querying.

Dedicated – you need to make a compute pool, you can optimize for performance, more expensive, use compute and storage, would pull the gold data into the vm storage, can make it in manage -> sql pools -> +

Serverless is mostly used for smaller work loads – uses built in pool (see in manage and SQL pools), only compute (data exists in datalake, we can only query

If you look under the linked tab you can see that data lake is already connected as per when we created it)

If you look in the data lake you can browse the folder, selecting on then (e.g. gold), you can then see the tables. From here you can directly query by right clicking and new sql script then Select TOP 100 rows, then select delta (since we know its that type). Click apply.

We can then see that it’s quite similar to databricks with the connect to option which connects to your relevant compute. Click run and a pane appears at the bottom with the clean data.

Let’s update this to create a view. Make sure to update use database to the gold\_db. Then run.

Go back to data workspace and under gold\_db -> views -> there will be dbo.address -> … -> New SQL script -> Select Top 100 rows. You may need to click the … at SQL database and then refresh. Running this query will show the address table, because it’s querying over the data lake any updates will be automatically reflected in the view.

dbo stands for Database Owner. It is a default schema provided by SQL Server. A schema is a namespace within a database that contains database objects such as tables, views, and stored procedures.

The dbo schema is often the default schema for objects created by the database owner (usually the user with administrative privileges on that database). It helps organize database objects and can also help manage permissions.

To create a view for each of these we can create a pipeline to do it dynamically. This time in synapse, not ADF.

Go to develop -> + -> import / new and add in sp\_CreateSQLServerlessView\_gold

NOTE:

* CREATE OR ALTER PROC is used to create a stored procedure or modify it if it already exists. In this case, the stored procedure is named CreateSQLServerlessView\_gold.
  + In SQL Server, a stored procedure is a set of SQL statements that are stored in the database and can be executed repeatedly
* The procedure takes one parameter, @ViewName, which is of type nvarchar(100). This parameter allows the user to specify the name of the view to be created or altered.
* Inside the procedure, a dynamic SQL statement is constructed, which creates or alters a view with the specified name (@ViewName). The view is based on a SELECT \* FROM OPENROWSET query that reads data from a Delta Lake format file stored in Azure Data Lake Storage.
* The EXEC (@statement) command executes the dynamic SQL statement contained in the @statement variable.

Now publish to save the script. Next step is to create pipeline that uses this procedure. Now create a linked service to connect to this serverless SQL db

Manage -> llinked services -> + -> search for azure sql db -> rename -> autoresolve integration runtime (since in azure) -> account selection choose manually -> for fully qualified domain in synapse resource tab go settings/properties/ scroll down to find serverless sql endpoint -> add db name -> system assigned managed identity authentication (uses your email addresses access to connect to the db) -> test connection -> create -> publish all

Now we can use this linked service to access the stored procedure from the pipeline

Go integrate -> + -> pipeline -> search for get metadata (this will get table names) -> rename -> settings -> new dataset -> search and select datalake -> binary -> rename -> select our new linked service -> file path browse, select root/gold/SalesLT

(if you get error “Forbidden”, its because your synapse doesn’t have relevant permissions. To resolve: go to rg -> sa -> iam -> add -> select storage blob data reader -> then in members select managed identity -> select members -> select synapse -> review and assign. Now go back and click on the browse icon and your folders will come up)

FYI

Managed Identity is an identity provided by Azure Active Directory (Azure AD) that allows Azure services to authenticate to each other securely, without needing to manage credentials explicitly.

Linked Service in Azure Synapse (or Azure Data Factory) is a configuration that defines the connection to an external data source or service.

ok so in my example I have synapse that is trying to create a pipeline that uses datalake service

since datalake is external it required a linked service, this linkedservice says use my email authentication

however datalake doesn't know to expect or to let synapse in, so I have to explicitly go and say to the data lake (storage account) to expect and allow synapse (via IAM role assignment, using the managed identity to identify)

ok so click ok then field list pops up under settings, select child items (gets all child items under the saleslt folder (via browse)). To see the output lets run debug and the output pain will automatically pop up at the bottom. Hover over the activity name and you’ll see an output icon – click this and youll see the child items are the folders with names.

Now that we have these, lets create a foreach activity that gets the table name and passes as a parameter to the stored procedure. So type foreach activity and drag onto pipeline, setting it as on success from get metadata. Rename -> click settings -> items -> dynamic content -> under activity outputs select get table names and add childitems at end to get the whole array into the foreach.

To assign what it does on each foreach loop, select the pen edit icon inside the foreach activity. Then search for stored procedure activity and add on to screen (Drag and drop). To get our stored procedure, go settings -> linked service (select SQL) -> select your stored procedure. Then select stored procedure parameters and click new, here we add in the ViewName param from the procedure with type string, value as @item().name

Now publish all and trigger now. If you view pipeline run you can see their done. If you go to data -> refresh db -> expand view, you can see their al created. This pipeline only needs to be re-run if the schema changes, not the data since it’s a view

# Data Reporting using PowerBI

The online version of powerbi only lets you interact with the report not create it from scratch so for that you need the desktop app, which only works on windows. So if you don’t have a windows you’re best bet is to try virtualize using some software as you cant even spin up a windows vm since you need a license key.

So lets open power bi -> blank report -> get data -> azure -> synapase analytics -> server end point (same way as before, serverless sql endpoint from properties) -> add db name -> import loads data completely to power bi / direct query it does load, so go load -> ok -> authenticate with Microsoft account and log in as you azure portal email since you have access in their -> connect

Can now see all diff views, check all and click load, Then can see on right the data

Quick bi overview, 3 tabs on left; visual/report, data, modelling

Report – built in visualisations

Data – see the actual data

Modelling – (most important) can automatically create this but not completely accurate as it guesses relationships based on columns

To create relationships either drag and drop or click manage relationships (better viewing if model looks confusing), e.g. manage relationship -> new relationship -> from customeraddress -> to address -> many to many -> 1:1 -> save -> close

The reason you do this is so that filtering in one table is recognized in another, say if you have 2 charts and you’re filtering one you want both to recognize the change

You want to create this based on fact and dimension tables

Fact tables contain quantitative data for analysis, while Dimension tables contain descriptive attributes related to the data in the Fact table.

So now go to report tab and create simple visual

Click card -> go to data and select product, you can see we have product id, we will use this to represent total number of products in the table, so go back to reports expand product and drag and drop product id into fields section. You can see result is no updated, you can click drop down and see we can change this card field based on requirements

Lets do another for total sales, lets go data and salesorderdetail and see that line total is the total sale price so go back to report and click card -> open salesorderdetail and drag line\_total to field, it should auto as sum

Next lets add a donut chart for inferred gender say, customers have a title through which we could infer gender for our customer analysis, so drag that into legend and value

Now let’s add a filter, so select slicer (filter) and in field lets drag in title, this way we can see how this affects the related table (as per our data model). We could do another for model category using the same way.

# Security and Governance using Entra ID

Go to azure portal -> rg -> iam -> role assignments, notice that the only person with permission is you. If you were in a company there’d be multiple possible many people with access and people will join leave etc. One way we could do this is by assigning each individual person with specific access, but that can be messy and cause a bunch of problems with churn. So instead we can create a security group, a logical grouping of individuals who require the same level of access, then we can assign that group specific permissions. One group could be for example a team and whenever someone joins or leave they get added to the group. So lets create on.

Go to entra id (previously active directory) -> add -> group -> make up a name -> add owners (use your main admin account as this person can manage the group) -> add members -> create. Now refresh and see the group. Note that you can also use enterprise applications as members, e.g. synapse instead of using managed identity.

If we go back to IAM on the resource group go add -> add role assignment \_> select a role assignment -> then go to members and click select members then select your group -> review and assign. The members can now access the group.

# End to end pipeline testing

Lets go to datafactory and add a trigger to run this integrated pipeline when new rows get added. So launch data factory -> click on pipeline -> trigger -> new -> type (we could use events / schedule), lets use schedule -> specify start date -> timezone -> recurrence every 1 day -> update advanced recurrence option execute time to when we want it to run (e.g. in 10 mins time because we will go update SQL) -> lets specify end for tomorrow -> ok -> ok -> publish all -> publish

Now lets update data source in SSMS. Open SSMS and click connect. Then lets open Databases -> AdventureWorks -> Tables -> lets update product\_id from Products, since this has a count in powerbi that we can see update easily. So select SalesLT.Product -> right click -> Script table as -> INSERT to -> new query editor and you’ll see the insert to format, replace the VALUES with something like:

('Product Name', 'PN-001', 'Red', 50.00, 100.00, 'M', 1.5, 3, 12, '2024-08-25', NULL, NULL, 0x, 'thumb.jpg', NEWID(), GETDATE())

Execute and a new row will be added to the Product table. Ctrl + s to then save this new query.

Now let’s go to PowerBI and take note of the number of items. Then back to the datafactory -> monitor -> pipeline runs and wait for the time the trigger goes off. You’ll know it has when triggered by changes to schedule from manual (and it’s the right time). Lets click on the pipeline and we can see it go through the steps automatically – once its done if we refresh the report in powerbi by clicking home -> refresh. Wait for the refresh (which connects to sql and pulls latest data and will update queries thus report), then you’ll see the count of products update.

To now delete all of this, lets delete the resource group (and then any AD groups you don’t need)