

Azure Data Factory

ETL-pipeline

! Viktigt

Det här dokumentet förutsätter att du skapat de nödvändiga tjänsterna i Azure som står i dokumentet `Azure_Services.pdf`

Det här är ett utkast. En mer utarbetad version kommer så fort jag hinner.

Översikt

I det här dokumentet går jag kort igenom:

- *Linked services*
- *Datasets*
- *Activities*
 - *Copy data*
 - *Data flow*
- *Pipelines*

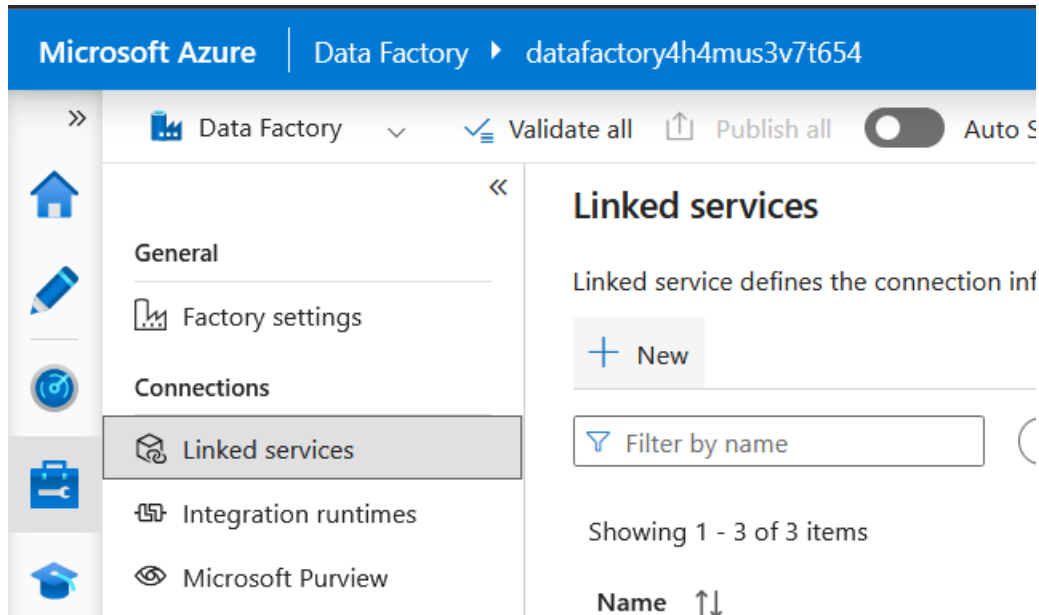
1 Linked services

I Azure Data Factory behöver du börja med att skapa *Linked services*. Det är kopplingar till dina tjänster i Azure.

Du behöver tre *Linked services*:

- En som heter `WWIRaw` och som hämtar rådatan från ett *Blob Storage*.
- En som heter `WWI` och som är kopplad till ditt eget *Blob Storage*.
- En som heter `WWICosmosDB` och som är kopplad till din Cosmos DB.

Du lägger till *Linked services* i *Manage*-fliken (se Figur 1). Klicka på + New.



Figur 1: Lägg till en *Linked service*, bild 1

1.1 Blob Storage

För att lägga till en *Blob Storage*, klicka på *Azure* och välj *Azure Blob Storage*.

För att få tillgång till rådatan:

- Ändra **Account selection method** till *Enter manually*
- skriv (eller klistra) in `storage4h4mus3v7t654` i rutan **Storage account name**
- kopiera och klistra in *account key* nedan i rutan **Storage account key**



För att skapa en *Linked service* till ditt eget *Blob storage* kan du använda *From Azure subscription* som **Account selection method** istället, först välja din *subscription* och sedan välja ditt *storage account* från listan.

1.1.1 Account key

3jy0DHZBNmmiw60iBeqikPwS3gxu4YK8r0g5axIswuOgd7K1ZbqYjg/9SyRKKdm6JnS0krb8rTl++AStCbzsmg==

Se Figur 2.


New linked service


 Azure Blob Storage [Learn more](#) 

Name *

WWIRaw

Description

Connect via integration runtime * 

 AutoResolveIntegrationRuntime

Authentication type

Account key

Connection string

Azure Key Vault

Account selection method 

☐ From Azure subscription ☒ Enter manually

Storage account name *

storage4h4mus3v7t654

Storage account key

Azure Key Vault

Storage account key *

Partitioned DNS enabled 

☐

Create

Back

 Test connection

Cancel

Figur 2: Lägg till en *Linked service*, bild 2

1.2 Cosmos DB

För att lägga till en *Linked service* till en Cosmos DB gör du ungefär likadant som med ett *Blob storage*. Använd *From Azure subscription* som **Account selection method** och välj din Cosmos DB i listan.

2 Datasets

Första steget i uppgiften är att skapa en *pipeline* som kopierar rådatan från **WWIRaw** till ditt eget *Blob storage*. För att göra det behöver du skapa *datasets*, som är referenser till **csv**-filerna.

2.1 Skapa *datasets*

I *Author*-fliken (pennan), Klicka på + och välj Dataset. Du kan också klicka på de tre punkterna brevid **Datasets** och välja *New dataset*.

I nästa steg ska vi ange att datan i vårt dataset kommer från ett *Blob storage*, så välj *Azure Blob Storage* och klicka *Continue*.

Datan kommer från **csv**-filer, så välj formatet *DelimitedText* och klicka *Continue*.

Det är viktigt att döpa datasetet så att vi kan identifiera det senare. Vi kan ändra namn senare, men det är en bra idé att göra det redan nu. Vi vet att vi ska läsa in tre dataset: **Customers**, **Orders** och **OrderLines**. Vi börjar med **Customers**, så skriv Customers i rutan under **Name**.

Sedan ska vi ange var datan kommer ifrån. Det är vår *Linked service* som vi döpte till *WWI-Raw*.

Till sist ska vi ange vilken fil i vårt *Blob storage* som innehåller datan. Klicka på mapp-ikonen till höger för att bläddra bland filerna. Eftersom vi ska lägga till vårt **Customers**-dataset är det filen **Sales.Customers.csv** vi ska leta fram. Den ligger i **wwi/raw/sales/**.

Klicka på **Sales.Customers.csv**, och sedan på OK, och på OK en gång till. Nu har du lagt till ett dataset! Du kan upprepa processen ovan för de två andra filerna, **Sales.Orders.csv** och **Sales.OrderLines.csv**.

2.2 Anpassa *datasets*

När du skapat dina datasets är nästa steg att se till att Data Factory kan läsa datan korrekt. Klicka på Customers-datasetet för att få fram det i Data Factory.

2.2.1 Separera kolumner

Nu kan vi förhandsgranska datan för att försäkra oss om att den läses in på ett korrekt sätt. Klicka på `Preview data`.

Det ser inte alls bra ut! All vår data är i en enda kolumn. Om vi tittar noga ser vi att kolumnerna är separerade med semikolon ; istället för det mer vanligt förekommande komma-tecknet ,. Vi behöver säga till Data Factory att dela upp datan med semikolon istället för komma. Det kan vi göra genom att välja semikolon som **Column delimiter**, eller genom att klicka på Detect format för att låta Data Factory själv avgöra vad den tycker verkar bäst. Oavsett metod, se till att det står Semicolon (;) i **Column delimiter**-rutan och klicka på `Preview data` igen.

Nu ser det bättre ut!

2.2.2 Saknade värden

Vi behöver också tala om för Data Factory vilket värde som används för att representera saknade värden i datan. Skriv NULL i rutan **Null value**.

2.2.3 Schema

Klicka på Schema-fliken. Vi ser att schemat inte är uppdaterat efter att vi bytte till semikolon. Klicka på Import schema och välj From connection/store. Nu är schemat uppdaterat och vårt dataset redo att användas!

Upprepa ovanstående steg för de återstående två dataseten.

3 Activities

3.1 Copy data

Nu är vi redo att skapa en *pipeline*! Den ska kopiera våra tre dataset från WWIRaw till ditt eget *Blob storage*.

Klicka på + igen, och välj Pipeline -> Pipeline. Ändra namn från pipeline1 till CopyData i **Properties**-rutan till höger.

Under **Activities**, klicka på **Move and transform** och dra in en Copy data till arbetsytan.

I **General**-fliken, byt namn till CopyCustomers.

3.1.1 Source

Vi behöver tala om vilken datakällan är, det vill säga varifrån datan kommer - det är vårt **Customers**-dataset.

Klicka på Source och välj Customers i **Source dataset**-rutan.

3.1.2 Sink

Vi behöver också ange vart datan ska ta vägen. Det gör vi i Sink-fliken.

Vi har inget dataset just nu, men vi kan skapa ett. Klicka på + New och välj Azure Blob Storage. Klicka på *Continue*.

Välj DelimitedText och klicka på *Continue*.

Döp det nya datasetet till CustomersCopy och välj den *Linked service* som pekar till ditt eget *Blob storage*.

I **File path**, skriv wwi i Container-rutan. Klicka på OK.

Ändra filändelsen i **File extension**-rutan från .txt till .csv.

3.1.3 Klona Copy data-steg

Nu har vi gjort ett Copy data-steg för Customers-datan. Vi kan ha flera Copy data-steg i samma *pipeline*. Istället för att gå igenom alla stegen ovan igen, kan vi klona vårt CopyCustomers-steg och ändra detaljerna.

Klicka på Clone. Nu har vi en kopia av vårt CopyCustomers-steg och kan ändra det så att det blir ett CopyOrders-steg istället.

Börja med att ändra namn från CopyCustomers_copy1 till CopyOrders. I Source-fliken, ändra **Source dataset** till Orders. I Sink-fliken, klicka på + New och repetera stegen från Avsnitt 3.1.2 ovan.

Gör sedan om processen en gång till, det vill säga klona ett av Copy data-stegen och anpassa det till OrderLines-datasetet.

När det är gjort, klicka på Publish all för att spara dina ändringar. Om du får felmeddelanden, läs dem och försök lösa problemet med hjälp av det som står i felmeddelandet.

3.1.4 Kör *pipeline*n

Nu är den första *pipeline*n klar! För att faktiskt köra den och kopiera datan behöver vi klicka på Add trigger och klicka på Trigger now.

Du kan följa processen i Monitor-fliken. När den är klar har du kopierat datan och är redo för nästa steg: ett *Data flow*!

3.2 Data flow

I vårt *Data flow* ska vi tvätta datan och slå ihop våra tre dataset till ett enda, som i slutet ska lagras i en CosmosDB-dokumentdatabas.

Klicka på + och välj Data flow -> Data flow.

Byt namn från dataflow1 till SalesETL.

3.2.1 Sources

Ett *data flow* fungerar ungefär som en *pipeline*, men används för att tvätta och transformera data. Vi ska lägga till våra kopierade dataset som källor, *Sources*.

Klicka på Add Source och välj Add Source.

Ändra **Output stream name** från source1 till Customers.

Som **Dataset**, välj CustomersCopy.

Gör om processen ovan med Orders och OrderLines så att vi har tre datakällor.

3.2.2 Ändra datatyper

I Projection-fliken kan vi ändra datatyper på kolumner. Det finns inga kolumner i Customers som vi behöver ändra, men väl i Orders och OrderLines.

Orders

Kolumn	Önskad datatyp	Format
OrderDate	date	dd/MM/yyyy

OrderLines

Kolumn	Önskad datatyp
Quantity	integer
UnitPrice	float

3.2.3 Filtera

Vi vill inte behålla alla kolumner i datan. Använd Select-transformern för att filtera datakällorna. Döp stegen till [NamnetPåKällan]Filtered, till exempel CustomersFiltered.

Från **Customers** ska du spara:

- CustomerID
- CustomerName

Från **Orders** ska du spara:

- OrderID
- CustomerID
- OrderDate

Från **OrderLines** ska du spara:

- OrderLineID
- OrderID
- StockItemID
- Description
- Quantity
- UnitPrice

3.2.4 Joins

Vi ska slå ihop våra tre dataset till ett. Det gör vi med *joins*. Första steget är att slå ihop OrderLines och Orders. Vi vill ha OrderLines till vänster i vår *join*, så koppla en Join till OrderLinesFiltered.

Välj OrdersFiltered som **Right stream** och joina på OrderID. Kalla transformationen för OrdersJoined.

Notera

För att kunna förhandsgranska datan i Data preview-fliken behöver du först ha aktiverat Data flow debug.

Koppla en Join till till OrdersJoined. Välj CustomersFiltered som **Right stream** och joina på CustomerID. Kalla steget för CustomersJoined.

3.2.5 Sink

Sista steget är att koppla en Sink för att tala om vart datan ska ta vägen efter den har gått igenom alla transformationerna.

Koppla en Sink till CustomersJoined.

I Sink-fliken, välj Inline som **Sink type** och välj Azure Cosmos DB for NoSQL som **Inline dataset type**.

Välj din *Linked service* till din CosmosDB som **Linked service**.

Nu är ditt *data flow* redo att stoppas in i en *pipeline*.

4 Pipelines

Skapa en ny Pipeline och döp den till ETL.

Från Move and transform, dra in ett Data flow.

I Settings-fliken, välj SalesETL som **Data flow**.

Nu kan du testa att köra din ETL-pipeline. Glöm inte att klicka på Publish all innan för att spara dina ändringar.