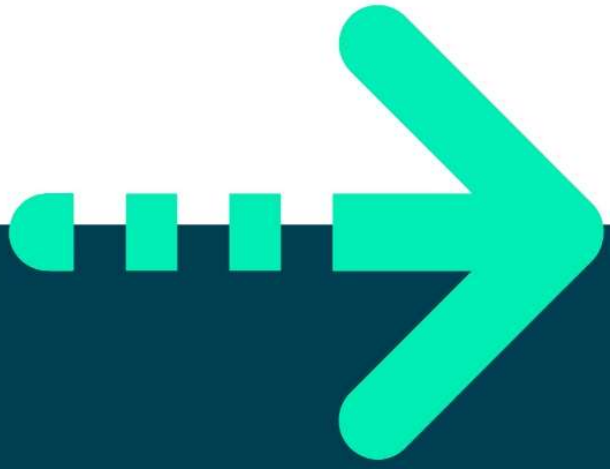
CREATING AN AZURE DATA WAREHOUSE

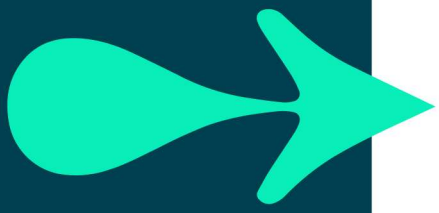
FOR ORDNANCE SURVEY



CONTENTS

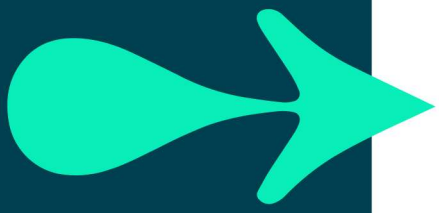

POWERING
POTENTIAL
Azure Data
Warehouse for
Ordnance Survey

Course Outcomes



- Understand how to design a data warehouse structure.
- Be able to create an Azure SQL database and create objects in the database.
- Use Azure Data Factory to move data into Azure Data Lake and to populate tables in Azure SQL Database.
- Be able to create Data Flows in Azure Data Factory.
- Use Stored Procedures to process data.
- Create and process Azure Analysis Services Data Model.
- Understand security and management issues with data in Azure.

Course Outline



1. Introduction to Data Warehousing
2. Designing a Data Warehouse
3. Implementing a Data Warehouse Structure with Azure SQL Database
4. Creating Azure Storage Accounts
5. Introduction to Azure Data Factory
6. Creating an ADF Transformation Data Flow
7. Creating an incremental Data Load
8. Introducing Azure Analysis Services
9. Security of Azure Data
10. Management of Azure Resources

Chapter 1 – Introduction to Data Warehousing

- What is a Data Warehouse?
- Why use a Data Warehouse?
- Data Warehouse Database and Storage
- Data Sources
- Extract, Transform, and Load Processes
- Light Weight Data Warehouse in OS

7

What is a Data Warehouse?

A centralized store of business data for reporting and analysis that typically:

- Contains large volumes of historical data
- Is optimized for querying, as opposed to inserting or updating data
- Is incrementally loaded with new business data at regular intervals
- Provides the basis for reporting solutions

Why use a Data Warehouse?

A successful business needs to be able to adapt — the following problems make that difficult:

- Business data is spread across many systems
- Data can be inconsistent, duplicated, and contradictory
- Fundamental questions can't be easily answered

Data Warehouse Database and Storage

- Database Schema
- Hardware
- High Availability and Disaster Recovery
- Security

10

Data Sources

Where is the data and how do we get it?

- Data Source Connection Types
- Credentials and Permissions
- Data Formats
- Data Acquisition Windows

Data quality

- Cleansing data
- Validating data values
- Ensuring data consistency
- Identifying missing values
- Deduplicating data

11

Extract, Transform, and Load Processes

Staging:

- What data must be staged?
- Staging data format

Required transformations:

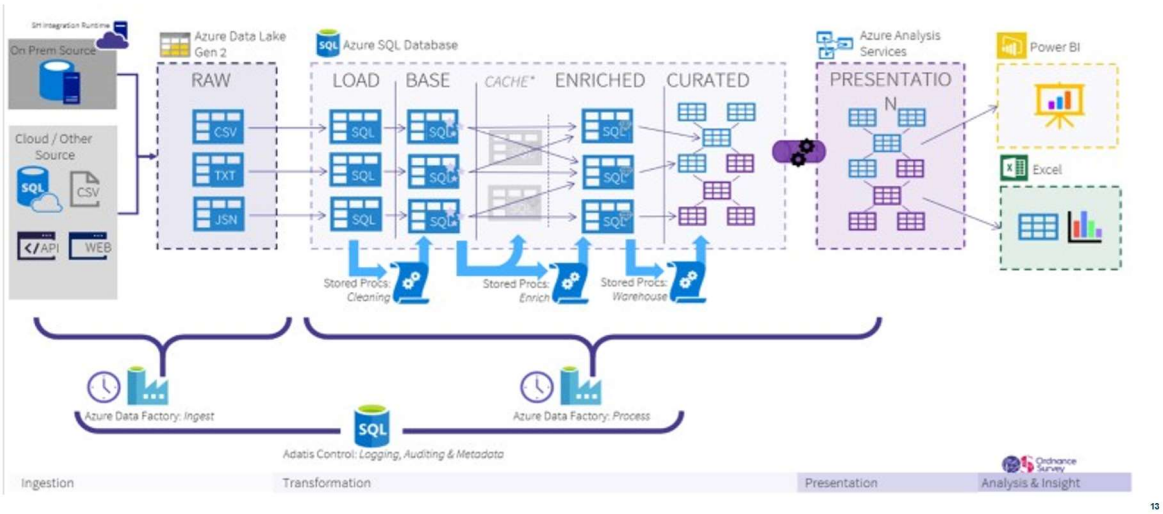
- Transformations during extraction versus data flow transformations

Incremental ETL:

- Identifying data changes for extraction
- Inserting or updating when loading

12

Light Weight Data Warehouse at OS



In this course we are focusing on the Azure technologies required to host and populate the data at various points in the data life-cycle.

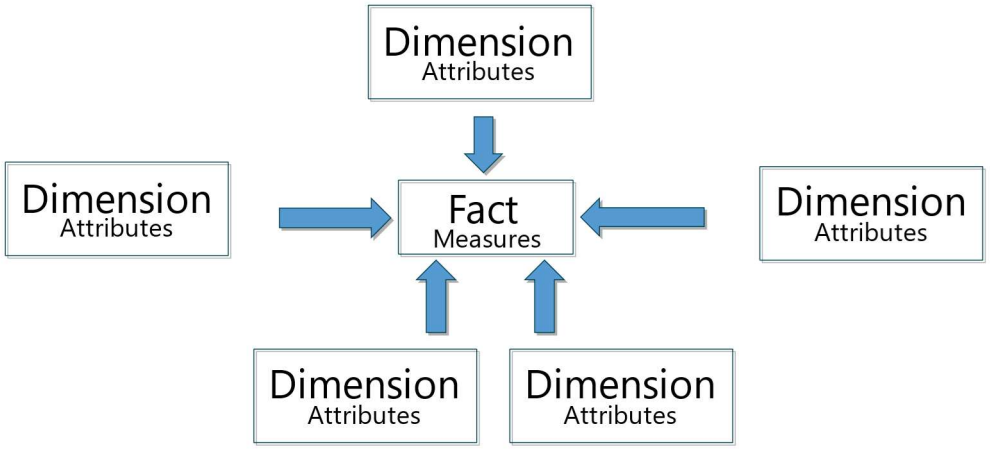
- Storage Accounts
- Data Lake Storage
- Azure SQL Database
- Azure Analysis Services
- Azure Data Factory

Chapter 2 – Designing a Data Warehouse

Data Warehouse Design Overview
Designing Dimension Tables
Designing Fact Tables

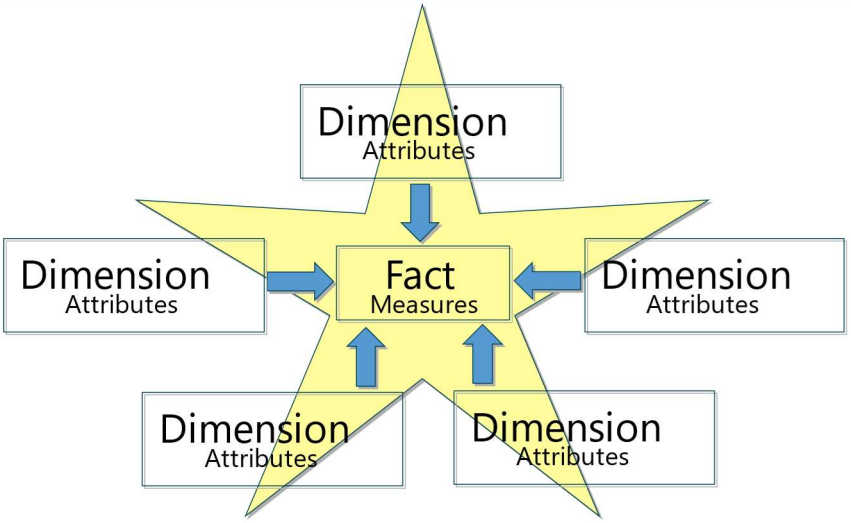
15

The Dimensional Model



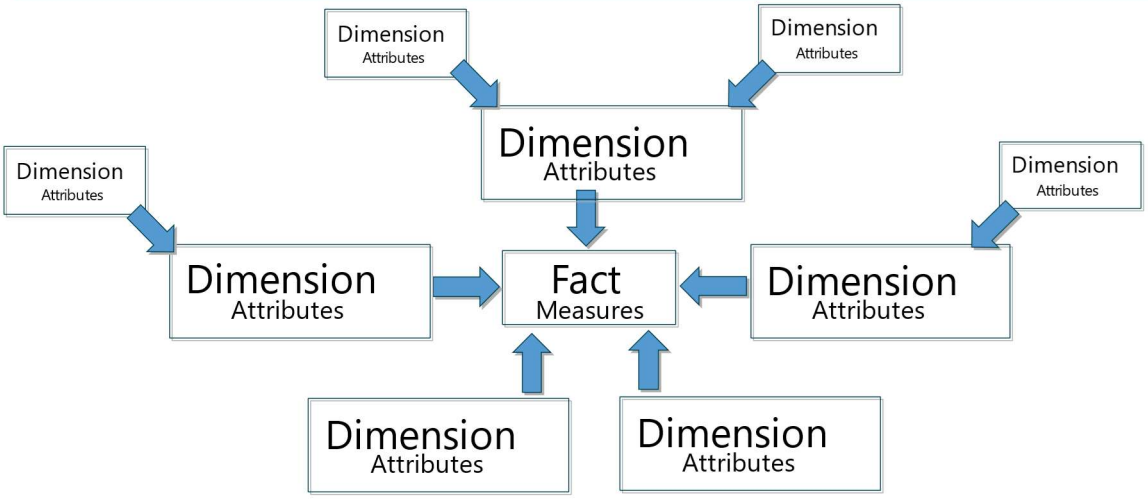
16

Star Schema



17

Snowflake Schema



18

Dimensional Model Design Process

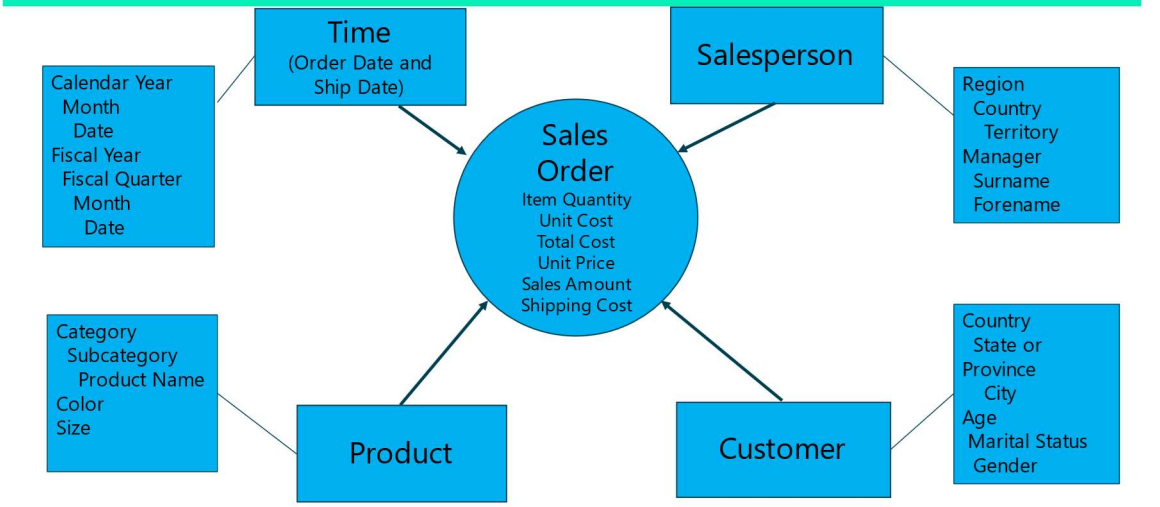
1. Identify the business process to model
2. Declare the grain* of the business process
3. Choose the dimensions that apply
4. Identify the numeric facts to populate the fact table

* The Grain is the level of detail represented by a row in the fact table

A Data Warehouse may consist of multiple Dimensional models with shared (conformed) dimensions.

19

Documenting Dimensional Models



20

Designing Dimension Tables

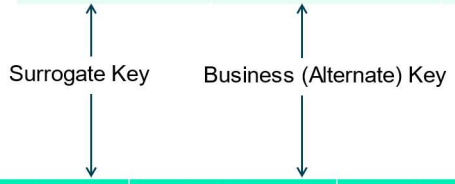
Design aspects to consider:

- Dimension Keys
- Dimension Attributes and Hierarchies
- Slowly Changing Dimensions
- Time Dimension Tables
- Self -Referencing Dimension Tables
- Junk Dimensions

21

Considerations for Dimension Keys

CustomerKey	CustomerAltKey	Name
1	1002	Amy Alberts
2	1005	Neil Black

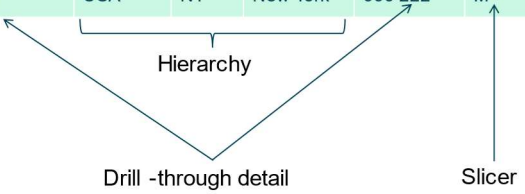


ProductKey	ProductAltKey	ProductName	Color	Size
1	MB1-B-32	MB1 Mountain Bike	Blue	32
2	MB1-R-32	MB1 Mountain Bike	Red	32

22

Dimension Attributes and Hierarchies

CustKey	CustAltKey	Name	Country	State	City	Phone	Gender
1	1002	Amy Alberts	Canada	BC	Vancouver	555 123	F
2	1005	Neil Black	USA	CA	Irvine	555 321	M
3	1006	Ye Xu	USA	NY	New York	555 222	M



23

Designing Slowly Changing Dimensions

CustKey	CustAltKey	Name	Phone
1	1002	Amy Alberts	555 123

Type 1

CustKey	CustAltKey	Name	Phone
1	1002	Amy Alberts	*555 222*

CustKey	CustAltKey	Name	City	Current	Start	End
1	1002	Amy Alberts	Vancouver	Yes	1/1/2000	

Type 2

CustKey	CustAltKey	Name	City	Current	Start	End
1	1002	Amy Alberts	Vancouver	*No*	1/1/2000	*1/1/2012*
4	*1002*	*Amy Alberts*	*Toronto*	*Yes*	*1/1/2012*	

CustKey	CustAltKey	Name	Cars
1	1002	Amy Alberts	0

Type 3

CustKey	CustAltKey	Name	*Prior Cars*	*Current Cars*
1	1002	Amy Alberts	*0*	*1*

24

Time Dimension Tables

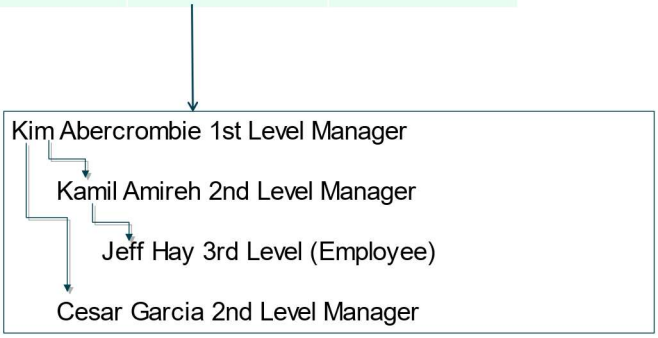
DateKey	DateAltKey	MonthDay	WeekDay	Day	MonthNo	Month	Year
00000000	01-01-1753	NULL	NULL	NULL	NULL	NULL	NULL
20160101	01-01-2016	1	3	Tue	01	Jan	2016
20160102	01-02-2016	2	4	Wed	01	Jan	2016
20160103	01-03-2016	3	5	Thu	01	Jan	2016
20160104	01-04-2016	4	6	Fri	01	Jan	2016

- Surrogate key
- Granularity
- Range
- Attributes and hierarchies
- Multiple calendars
- Unknown values

25

Self-Referencing Dimension Tables

EmployeeKey	EmployeeAltKey	EmployeeName	ManagerKey
1	1000	Kim Abercrombie	NULL
2	1001	Kamil Amireh	1
3	1002	Cesar Garcia	1
4	1003	Jeff Hay	2



26

Junk Dimensions

JunkKey	OutOfStockFlag	FreeShippingFlag	CreditOrDebit
1	1	1	Credit
2	1	1	Debit
3	1	0	Credit
4	1	0	Debit
5	0	1	Credit
6	0	1	Debit
7	0	0	Credit
8	0	0	Debit

- Combine low -cardinality attributes that don't belong in existing dimensions into a junk dimension
- Avoids creating many small dimension tables

27

Designing Fact Tables

Considerations when creating Fact Tables:

- Fact Table Columns
- Types of Measure
- Types of Fact Table

28

Fact Table Columns

Dimension Keys

→ Relates to Primary key in the Dimension tables

OrderDateKey	*ProductKey*	*CustomerKey*	OrderNo	Qty	SalesAmount
20160101	***25***	***120***	1000	1	350.99
20160101	***99***	***120***	1000	2	6.98
20160101	***25***	***178***	1001	2	701.98

Measures

→ The numerical values used for calculations

OrderDateKey	ProductKey	CustomerKey	OrderNo	*Qty*	*SalesAmount*
20160101	25	120	1000	***1***	***350.99***
20160101	99	120	1000	***2***	***6.98***
20160101	25	178	1001	***2***	***701.98***

Degenerate Dimensions

→ Values for each row stored in the Fact table

OrderDateKey	ProductKey	CustomerKey	*OrderNo*	Qty	SalesAmount
20160101	25	120	***1000***	1	350.99
20160101	99	120	***1000***	2	6.98
20160101	25	178	***1001***	2	701.98

29

Types of Measure

Additive

→ Measures can be aggregated at all dimension levels

→ Most Common type

OrderDateKey	ProductKey	CustomerKey	SalesAmount
20160101	25	120	350.99
20160101	99	120	6.98
20160102	25	178	701.98

Semi -Additive

→ Measures can only be aggregated on some dimensions

→ E.g. Stock levels can not be added day to day.

DateKey	ProductKey	StockCount
20160101	25	23
20160101	99	118
20160102	25	22

Non -Additive

→ Can not be aggregated on any dimension

OrderDateKey	ProductKey	CustomerKey	ProfitMargin
20160101	25	120	25
20160101	99	120	22
20160102	25	178	27

30

Types of Fact Table

Transaction Fact Table

OrderDateKey	ProductKey	CustomerKey	OrderNo	Qty	Cost	SalesAmount
20160101	25	120	1000	1	125.00	350.99
20160101	99	120	1000	2	2.50	6.98
20160101	25	178	1001	2	250.00	701.98

Period Snapshot Fact Table

State of data at a point in time

DateKey	ProductKey	OpeningStock	UnitsIn	UnitsOut	ClosingStock
20160101	25	25	1	3	23
20160101	99	120	0	2	118

Accumulating Snapshot Fact Table

Rows are updated as data changes

OrderNo	OrderDateKey	ShipDateKey	DeliveryDateKey
1000	20160101	20160102	20160105
1001	20160101	20160102	00000000
1002	20160102	00000000	00000000

31

Adventure Works is wanting to improve their reporting. These are some of the requests that have been made for reports:

Total Sales and profit by year, quarter and month.

Top 10 selling products by total sales and by quantity.

Which are the most profitable products?

Top 10 Resellers in each country.

Comparison of Total sales by salesperson against their target.

Can we get a map showing sales across the world?

Using SQL Server Management Studio, look at the source database (the instructor will give the location and connection information) and files in the data folder. We are only going to look at the Reseller related data. Document your ideas in a Word document.

1. Identify appropriate Dimension tables.
2. Identify an appropriate Fact table.
3. Identify the measures in the fact table.
4. Identify attributes and hierarchies for the Dimensions.
5. List any other points of note.

Be ready to share your ideas with the group.

Summary

A Data Warehouse often follows a Star Schema Design

Fact Tables contain the numerical values used in calculations

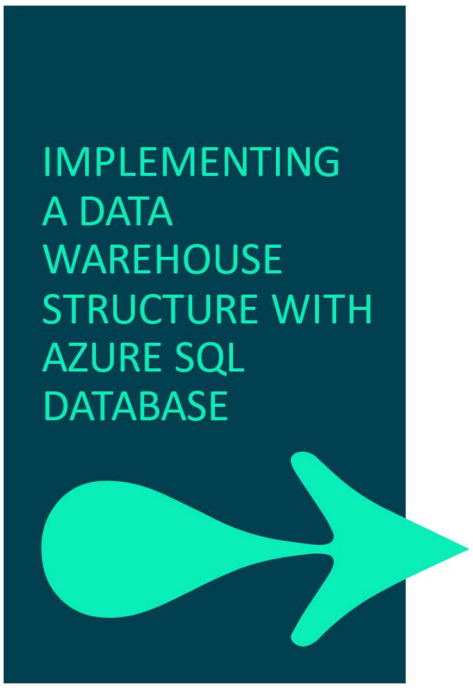
Dimension Tables provide the attributes to give meaning to the measures

Consider the Grain of the Fact table

Consider the need to track changing data

32

Chapter 3 – Implementing a Data Warehouse Structure with Azure SQL Database



SQL Database options in Azure
Creating an Azure SQL Database
Accessing an Azure SQL Database
Azure SQL Database limitations
Creating Database objects
Indexes on Data Warehouse tables
Data Compression

34

SQL Database options in Azure

			
SQL Virtual Machines	SQL managed instance	SQL database	Elastic Pool
Full control over SQL Server instance and underlying OS (windows or linux)	Full SQL Server access and feature compatibility	Great for modern, cloud born applications. Fully managed	Cost effective solution for managing multiple databases
IAAS	PAAS	PAAS	PAAS

35

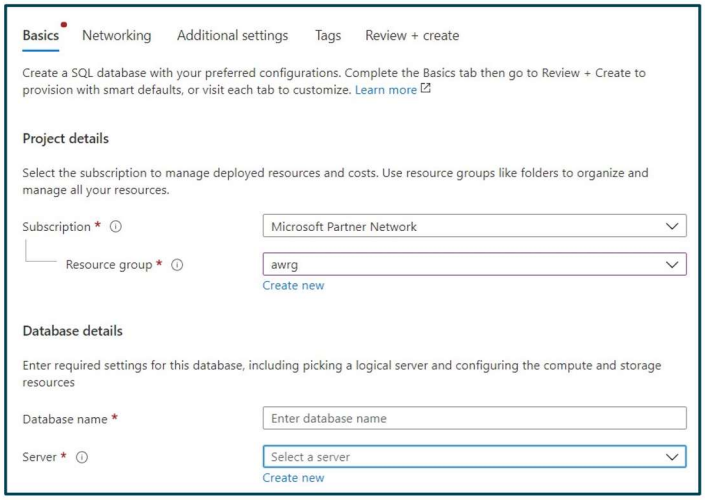
Creating an Azure SQL Database

Required settings:

- Subscription
- Resource Group
- Database Name
- Logical Server
- Compute and storage sizing

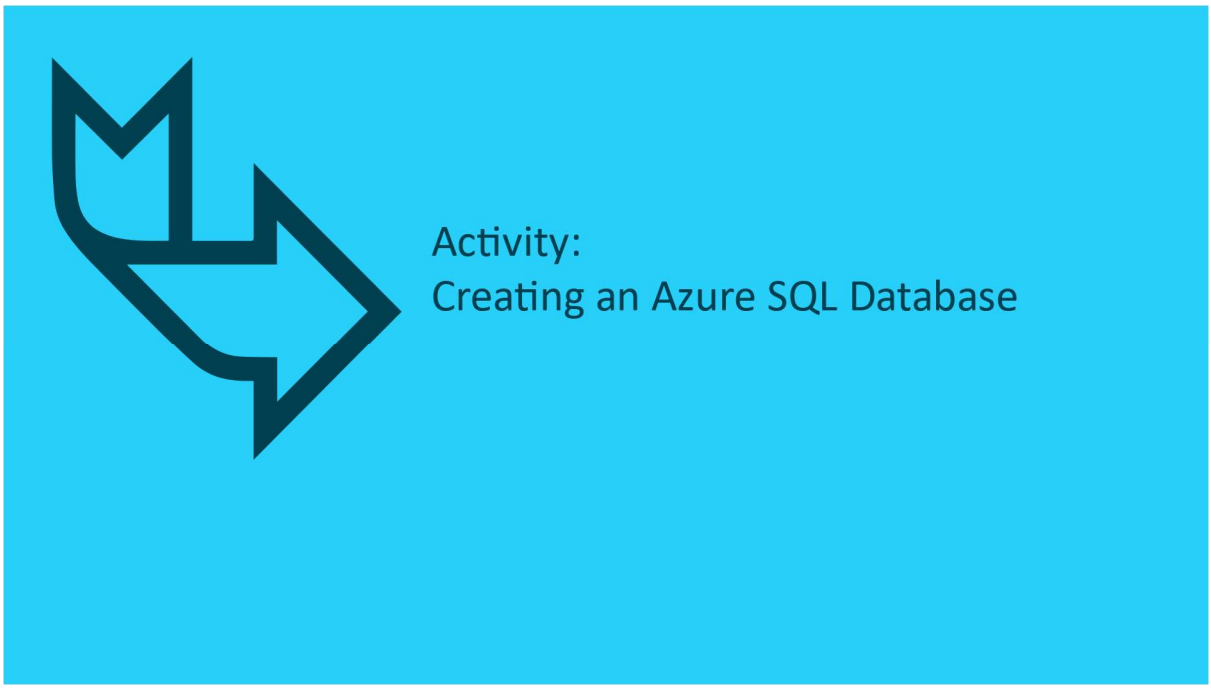
Optional settings:

- Network access
- Restore from backup or use sample data
- Tags for classification of resources



Demonstration

Creating an Azure SQL Database



In your Azure Subscription:

Create a Resource Group called ***osdwrg*** in the *UK South* region. (Click the *Create a resource* icon, search for Resource Group, Click *Create*, fill in the details. Click *Review+create*

From now on create all resources in the same Subscription, Resource Group and Region. Some resources need to have a globally unique name. These names will have xxx at the end. Replace the xxx with your initials and if necessary, a number to make a unique name.

Create an Azure SQL Database (search for SQL Database)
Database Name: ***ossqldbxxx***
Click *Create new* under server
Server Name: ***ossqlserverxxx***
Server admin Login: ***SQLAdmin***
Password: ***Pa55w.rd***
Compute+Storage: Click *Configure Database* and select ***Standard (S0)***
View the other settings for *Network* and *Additional Settings*, but don't make any changes.

Click ***Review+Create*** Check the settings

Click ***Create***

Once the Azure SQL Database has been created, look through all the configuration settings in the Azure Portal to familiarise yourself with the settings available.

At the end of this activity you should have created:

A Resource Group

A Logical SQL Server

An Azure SQL Database

Accessing an Azure SQL Database

Network access is controlled by Firewall Rules.

→ Add a rule to allow access from devices needing to manage and use the logical server

Admin User defined for the logical server at creation time.

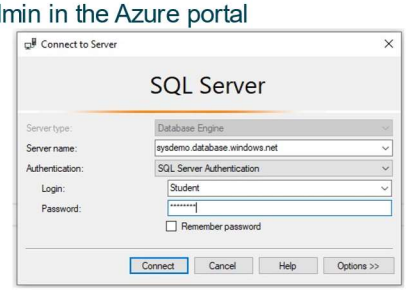
→ Password can be reset in the Azure portal

→ Azure Active Directory user or group can be added as an admin in the Azure portal

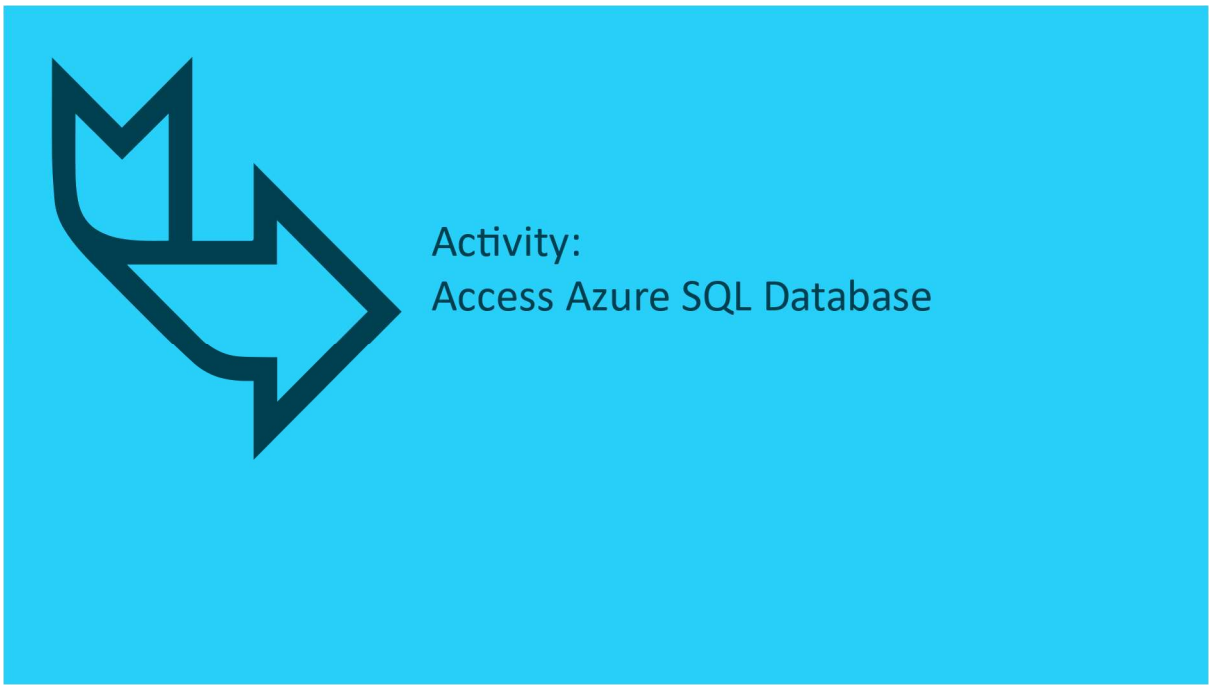
All other logins and user access is defined using SQL in the database

Connect using full dns name

e.g. sysdemo.database.windows.net



39



1. Locate your Azure SQL Server name (ossqlserverxxx.database.windows.net) and copy from the portal.
2. Open SQL Server Management Studio (SSMS) from the VM taskbar
3. Paste the server name into the Connection window.
4. Complete the other login details:
   Authentication: *SQL Server Authentication*
   Login: *SQLAdmin*
   Password: *Pa55w.rd*
5. Click the link in the error message to add a firewall rule for your IP address. (Login to Azure if necessary)
6. Check that you can see the database you have created – there are currently no objects in the database.

Azure SQL Database limitations

Queries that span multiple databases can not be used by default

- Elastic queries provide a possible solution
- Cross database queries are supported on managed instances and virtual machines

No support for common language runtime (CLR)

Files and File Groups restricted to Primary File Group

Database and Transaction Log backups are managed automatically

Functionality outside the scope of the database is not supported

41

Azure SQL Feature comparison:
https://docs.microsoft.com/en-us/azure/azure-sql/database/features-comparison

Elastic Queries – A method to allow cross data queries:
https://docs.microsoft.com/en-us/azure/azure-sql/database/elastic-query-overview#why-use-elastic-queries

Creating Database objects

Do not include USE *databasename* at the beginning of a script.

- Add a comment to indicate the database context.
- Check that you are running the script in the correct database

Check command syntax is appropriate for Azure SQL Database

- Select Azure SQL Database in the version selector in the documentation

Creating Tables and views is possible using graphical tools. (scripts are preferred)

Creating most objects from the Object Explorer generates a script template

- Ctrl + Shift + M op  ; the Specify Values for Template Parameters dialog

42

Indexes on Data Warehouse tables

Dimension table indexes

- Clustered index on surrogate key column
- Nonclustered index on business key and SCD columns
- Nonclustered indexes on frequently searched columns

Fact table indexes

- Clustered index on most commonly searched date key
- Nonclustered indexes on other dimension keys

  Or
- Columnstore index on all columns
- Composite index key comprises up to 16 columns

n.b. Columnstore indexes are not available in Azure SQL Database below S3 Pricing tier because of memory restrictions.

43

More about Column store indexes

https://azure.microsoft.com/en-gb/blog/columnstore-support-in-standard-tier-azure-sql-databases/

Data Compression

- Page compression increases the amount of data stored in each page
- Reduces the number of pages that need to be read for a query.
- Use page compression on all dimension tables, indexes and fact tables

n.b. Compression is not required with column store indexes.

44

Summary

- Create an Azure SQL Server
- Create Azure SQL Database
- Configure Security and Network access
- Create database objects
- Check the security access is appropriate
- Check the Database performance/cost is appropriate

1. Using SQL Server Management Studio, create the schemas that will be required.
    - load
    - stage
    - dw

Create a Database login and user that can be used for data load operations. It will require dbowner permissions. (e.g. *dataloader*)

Create a Database login and user with read access to data in the dw schema. This will be used for reporting. (e.g. *reportuser*)

Chapter 4 – Storage Accounts

Benefits of using Azure to store data

Comparing Azure to on- premises storage

Storage accounts

Storage account settings

Azure Data Lake Storage – Generation II

Create a Azure Data Lake Store (Gen II) using the portal

Compare Azure Blob Storage and Data Lake Store Gen 2

Uploading data with Azure Storage Explorer

49

Benefits of using Azure to store data

Automated backup

Multiple data types

Global replication

Support for data analytics

Encryption capabilities

Storage tiers

Comparing Azure to on-premises storage

The term “on-premises” refers to the storage and maintenance of data on local hardware and servers

Cost effectiveness	Reliability	Storage types	Agility
On-premises storage requires up-front expenses. Azure data storage provides a pay-as-you-go pricing model	Azure data storage provides backup, load balancing, disaster recovery, and data replication to ensure safety and high availability. This capability requires significant investment with on-premises solutions	Azure data storage provides a variety of different storage options including distributed access and tiered storage	Azure data storage gives you the flexibility to create new services in minutes and allows you to change storage back-ends quickly

Storage accounts

What is a storage account?

It is a container that groups a set of Azure Storage services. Only data services can be included in a storage account such as *Azure Blobs, Azure Files, Azure Queues, and Azure Tables*

How many do you need?

The number of storage accounts you need is typically determined by your data diversity, cost sensitivity, and tolerance for management overhead

The number of storage accounts you need is based on:

Data diversity:

Organizations often generate data that differs in where it is consumed and how sensitive it is

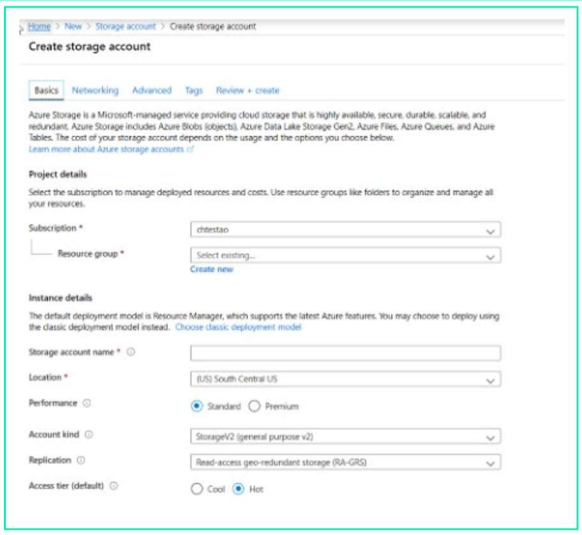
Cost sensitivity:

The settings you choose for the account do influence the cost of services, and the number of accounts you create

Management overhead:

Each storage account requires some time and attention from an administrator to create and maintain

Storage account settings



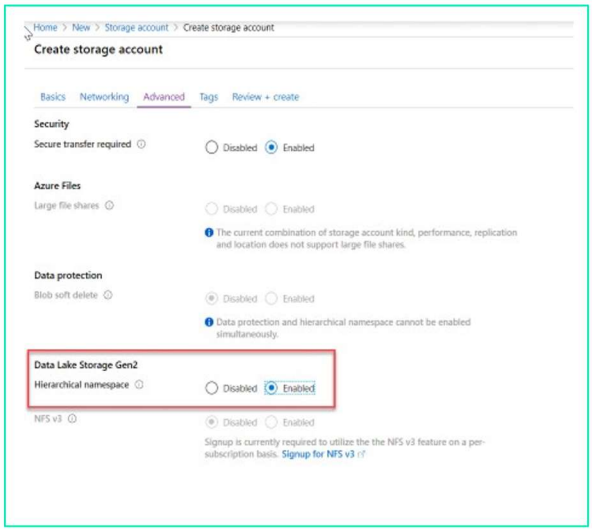
Create a Storage account in your subscription called ***ossageneralxxx***

Create a container called ***data***

Azure Data Lake Storage – Generation II

“Hadoop Access” is a key differentiator for using Azure Data Lake Storage Gen2 (ADLSG2) over Blob storage. Coupled with the security model that provides very granular permissions, this makes ADLSG2 a compelling data storage tier for a range of Azure data platform technologies.

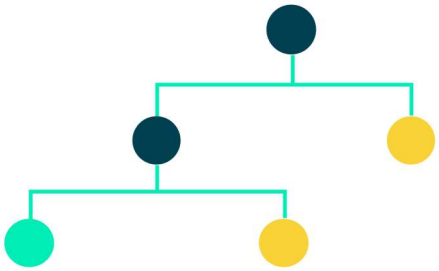
Create a Azure Data Lake Store (Gen II) using the portal



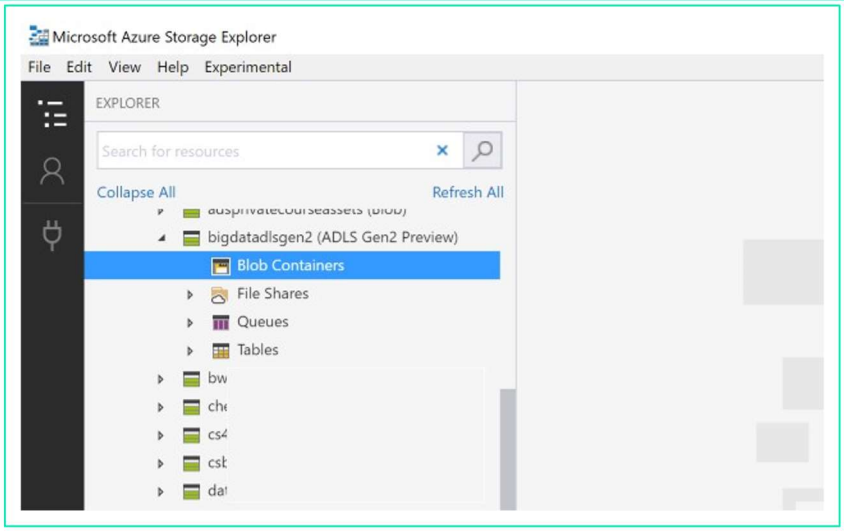
Create a Data Lake Gen 2 Storage Account call ***osdatalakexxx***

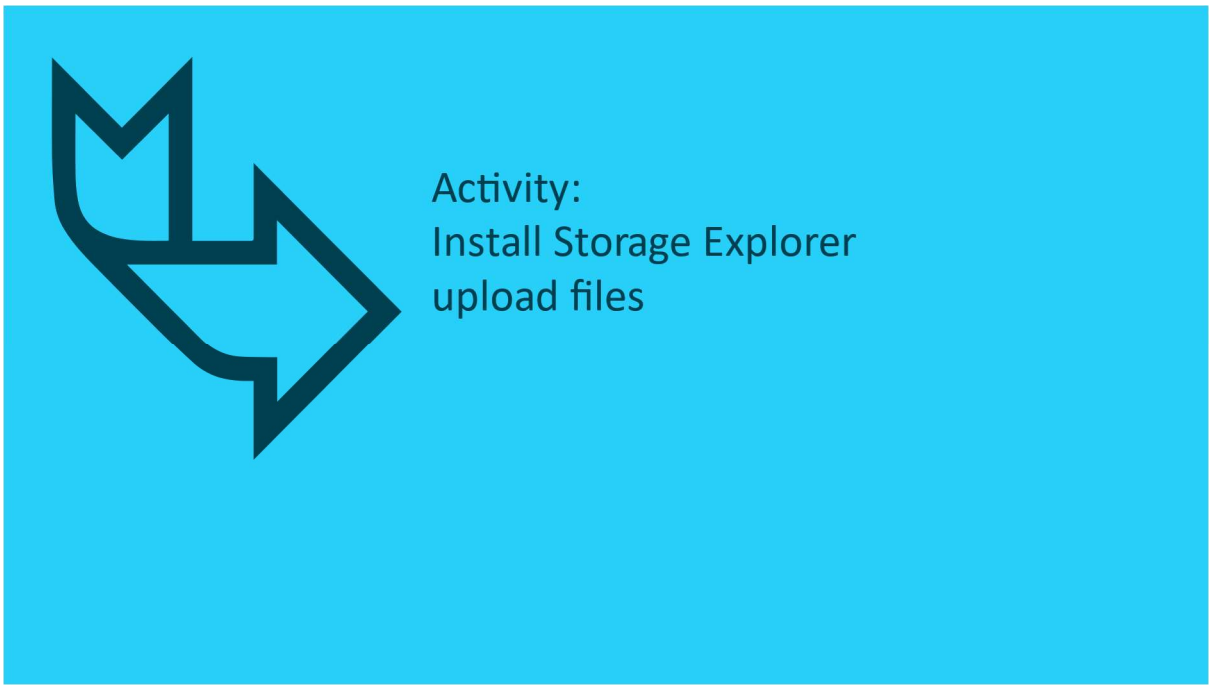
Create a container called ***data***.

Compare Azure Blob Storage and Data Lake Store Gen 2

Azure Blob Flat namespace	Data Lake (Gen II) Hierarchical namespace
	

Uploading data with Azure Storage Explorer





1. In your Data Lake Storage Account, click on *Overview*.

At the top, click on *Open in Explorer*.

Use the link in the portal to install Azure Storage Explorer in your Virtual Machine.

Copy the files from the Data folder in your VM to the data container.

Accessing Storage Accounts

Access to Storage Accounts and Data Lake Storage requires credentials and network access

Credentials:

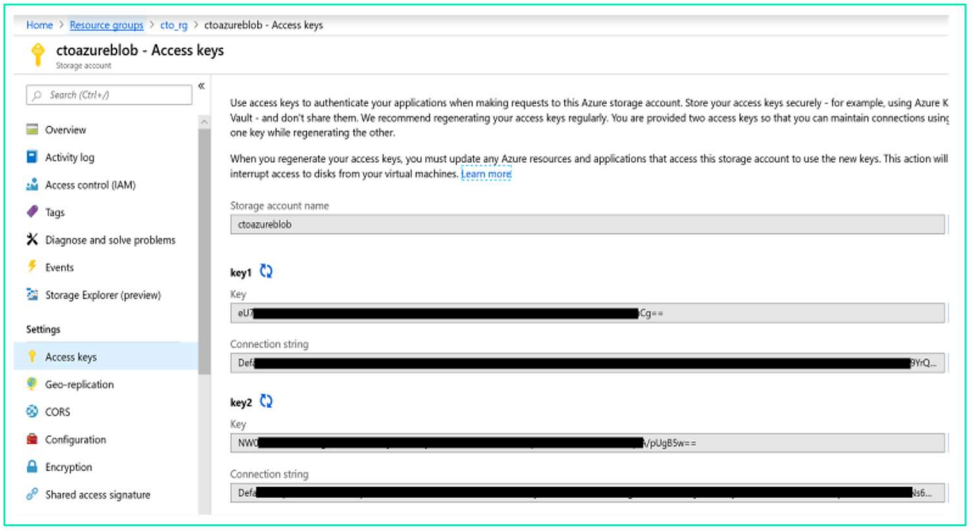
- Access Key
- Shared Access Signature
- Role Based Access Control (RBAC)

Network Access

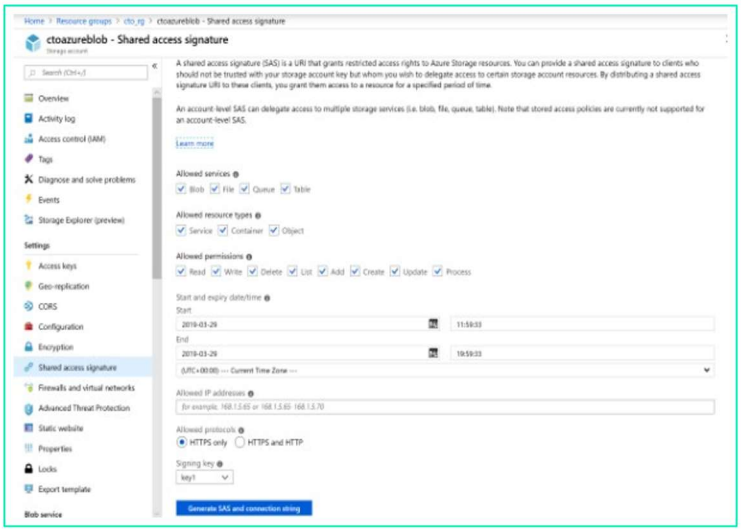
- Access to upload data
- Azure Data Factory Access

61

Storage account keys



Shared access signatures

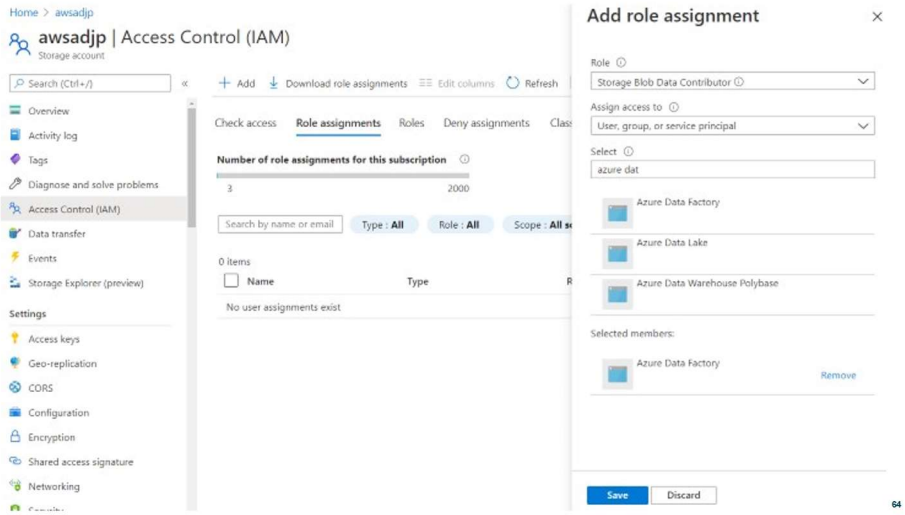


Role Based Access Control

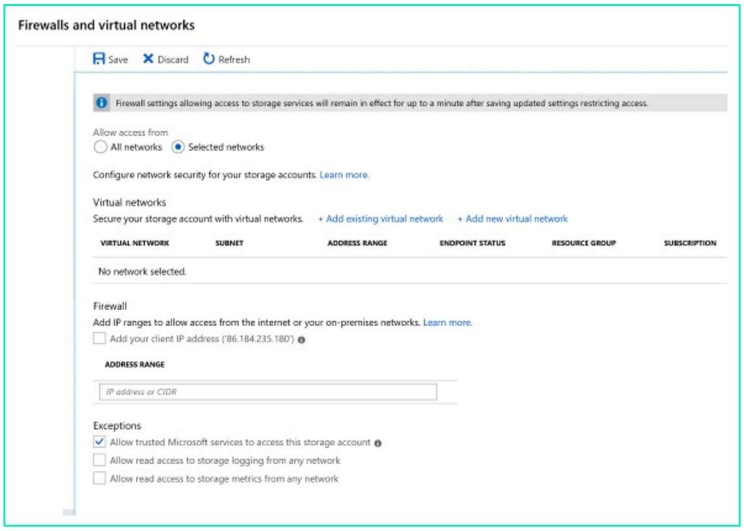
Select the Role

In the Select box, start to type the name of the user, group or service.

Click on one required and click Save



Control network access to data



Summary

Choose appropriate type of Storage Account

- Storage Account
- Data Lake Storage Gen 2.

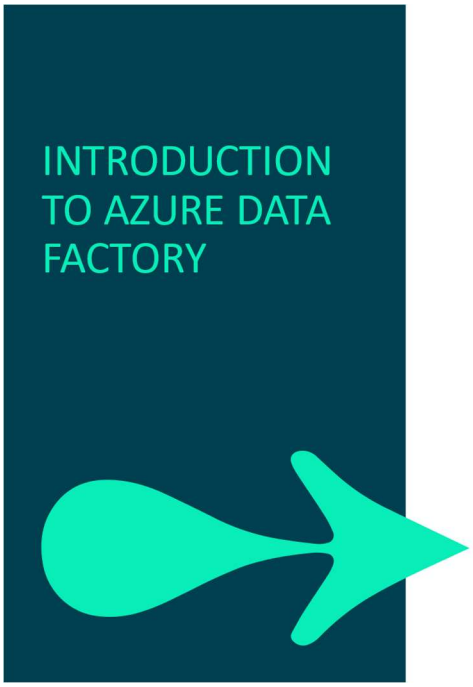
Create accounts and configure security

Create containers, folders etc

Load data

Check the security is appropriate

Chapter 5 – Azure Data Factory



What is Azure Data Factory?
Creating a Data Factory
Data Factory Components
Data Factory Integration Runtime

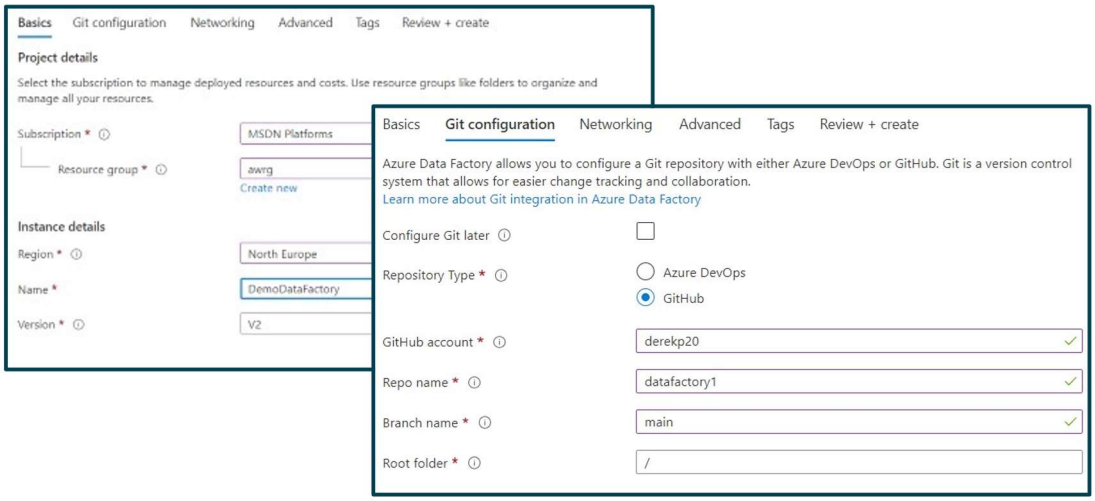
68

What is Azure Data Factory?

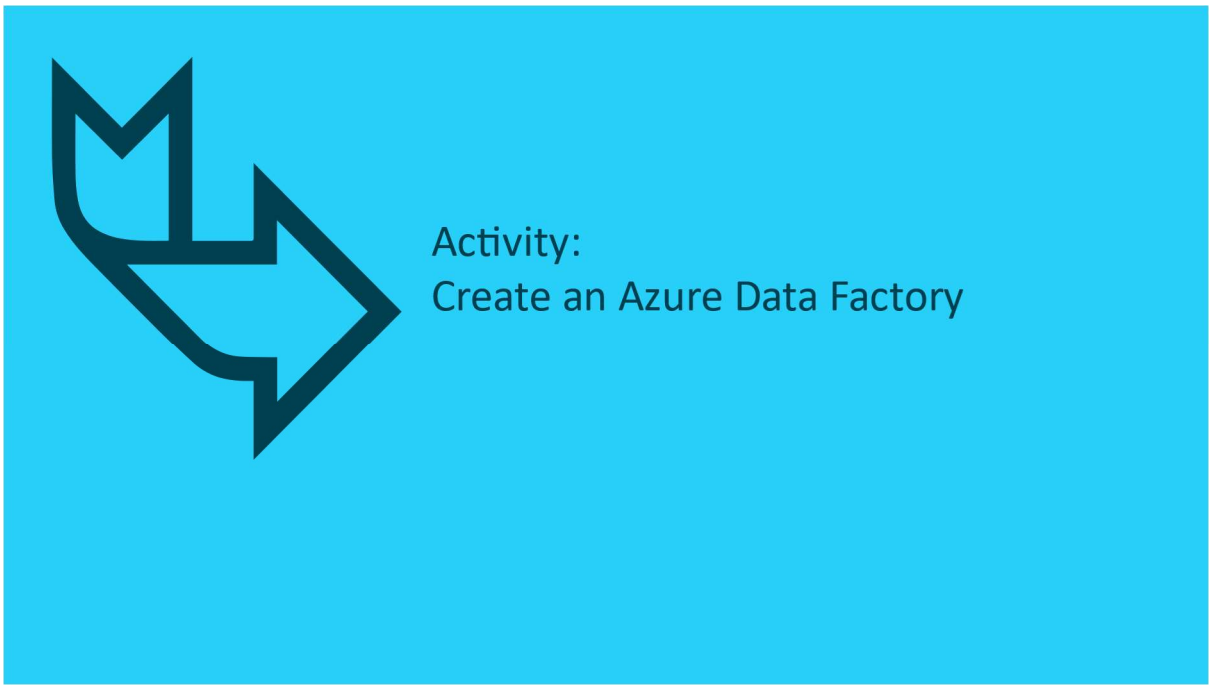
Creates, orchestrates, and automates the movement, transformation and/or analysis of data through in the cloud

69

Creating a Data Factory



70



Create an Azure Data Factory called ***osadfxxx***

Use the repository information that your instructor will give you.

Data Factory Components

Linked services

Datasets

Data Factory activities

Pipelines

72

Components – Linked Services

A Linked Service defines a connection to a data source or service

A Linked Service can be create by clicking on the Manage icon 
or when configuring an activity in a pipeline

e.g. Storage Account
Azure SQL Database
Databricks Service
HDInsight

Definition includes the connection string and credentials

73

If possible, create Linked Services first. They are then available as you start to create pipelines.

https://docs.microsoft.com/en-us/azure/data-factory/concepts-linked-services

Components – Dataset

Defines data to be accessed in a linked service

Requires the Linked Service to be created first

e.g. File in a container on a storage account

table in a SQL Database

Includes

- The location of the data – table/folder/file
- the structure – names and data type of fields
- Other settings to determine how to retrieve the data

74

https://docs.microsoft.com/en-us/azure/data-factory/concepts-datasets-linked-services

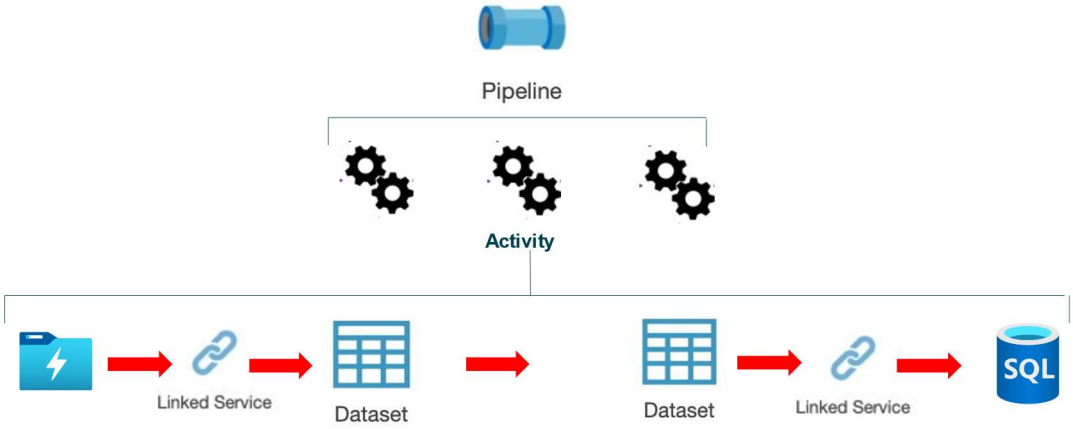
Components – Data Factory Activities

Data movement activities	Data movement activities simply move data from one data store to another. A common example of this is in using the Copy Activity
Data transformation activities	Data transformation activities use compute resource to change or enhance data through transformation, or it can call a compute resource to perform an analysis of the data
Control Activities	Control flow orchestrate pipeline activities that includes chaining activities in a sequence, branching, defining parameters at the pipeline level, and passing arguments while invoking the pipeline on -demand or from a trigger

75

https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities

Components - Pipeline



76

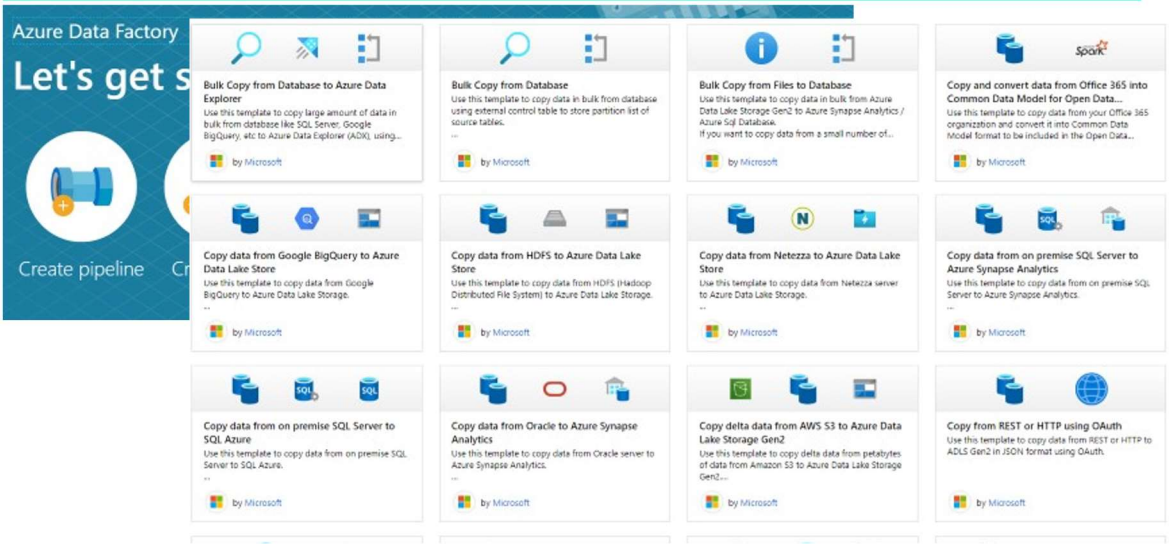
Data Factory Integration Runtime

IR type	Operation	Notes
Azure	Data Flow Data movement Activity dispatch	Default type used for data in the Azure environment
Self-hosted	Data movement Activity dispatch	Access to ON-premise data
Azure -SSIS	SSIS package execution	Runs SSIS packages in an Azure VM

77

https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime

Data Factory Wizards and Templates



1. Use the Copy option from the Getting Started page to copy the ***Targets*** file from the Data Lake into a new table called ***load.Targets*** in your Azure SQL Database.

   Allow the wizard to create the table, define the columns and data types.

2. Examine the objects that have been created:

   - Pipeline
   - Activity
   - Datasets
   - Linked Services

3. For each item, look at the various property tabs in the area in the lower part of the screen.
4. Create a new Linked Service for the source Adventureworks2020 database. Use the credentials previously used.

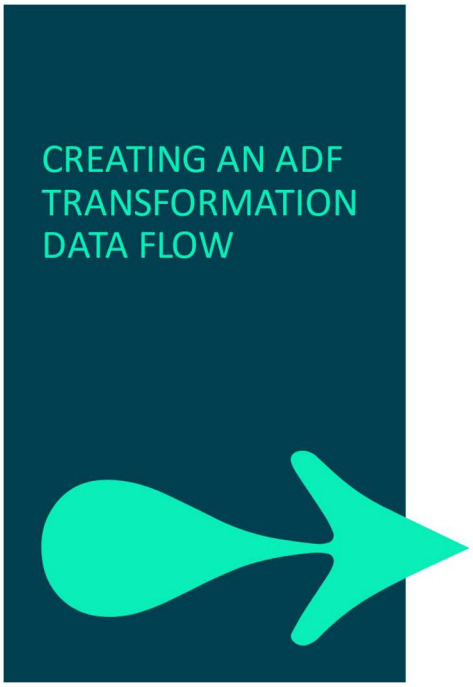
Orchestrates data movement and processing activities through Azure

Can work with many Azure Data Services

Wizards and templates make tasks easier.

80

Chapter 6 - Creating an ADF Transformation Data Flow



Creating a Transformation Data Flow in ADF

- Source
- Transformations
- Expressions
- Sinks

Transforming Data with SQL Stored Procedures

Parameterizing Pipelines

82

Creating a Transformation Data Flow

A Mapping Data Flow is a visual designer for data transformations
Added as an Activity in a Pipeline
Data flow debug should be enabled to view the changes through the design
Transformations are added as a sequence from a source to a sink
There can be multiple sources and multiple sinks
Usually sources are on the left moving to sinks on the right of the design surface
Each transformation has configuration settings in the lower part of the screen
The Data Preview tab enables viewing the results of each transformation in debug

n.b. The data flow can not be saved if incomplete unless using a Code Repository

83

Data Flow Overview:

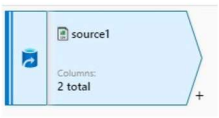
https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-overview

Data Flow Debug Mode:

https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-debug-mode

Data Flow Source

Source Type normally a Dataset, but can be other sources
Choose from existing datasets, or create a new one
Schema Drift allows for changes in the source columns



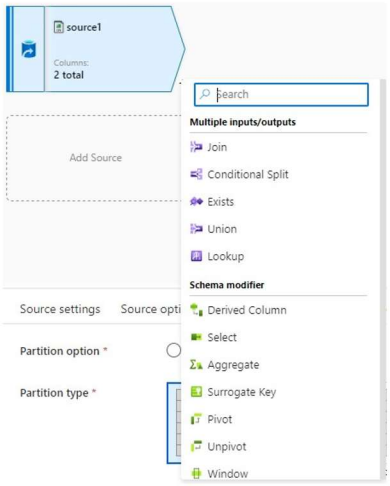
Source Options vary depending on the type of source
Projection used with file sources to define column data types
Optimize contains settings to configure partitioning schemes
Inspect provides a view into the metadata of the data stream that you're transforming
Data Preview allows a view of the data after the transformation

84

Schema Drift:

https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-schema-drift

Data Flow Transformations



Multiple inputs/outputs

- Join
- Conditional Split
- Exists
- Union
- Lookup

Schema modifier

- Derived Column
- Select
- Aggregate
- Surrogate Key
- Pivot
- Unpivot
- Window
- Flatten
- Rank

Row modifier

- Filter
- Sort
- Alter Row

85

https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-manage-graph

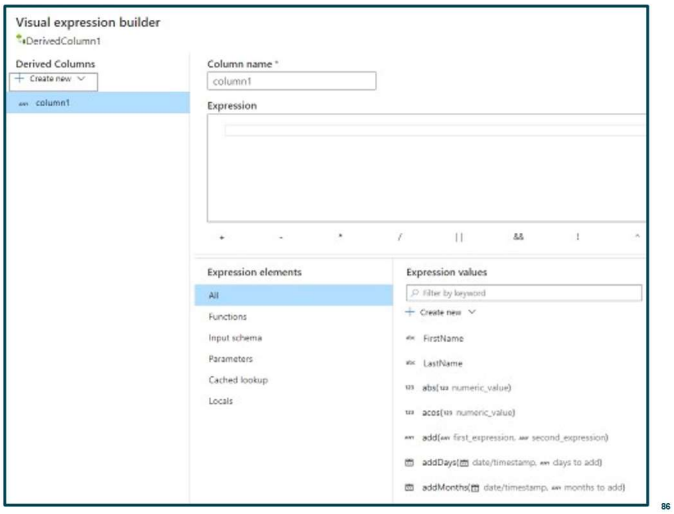
Expression Builder

The Expression Builder is used in a number of transformations to define values

Expressions can be built from

- input columns
- Functions
- Parameters
- Cached Lookup
- Local variables

Intellisense guides the creation of expressions



Using the Expression Builder:

https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-expression-builder

Data Flow Language Reference:

https://docs.microsoft.com/en-us/azure/data-factory/data-flow-expression-functions

Data Flow Sink

After the transformations, data is written to a destination store.

A sink is always required

There can be multiple sinks

A sink can be

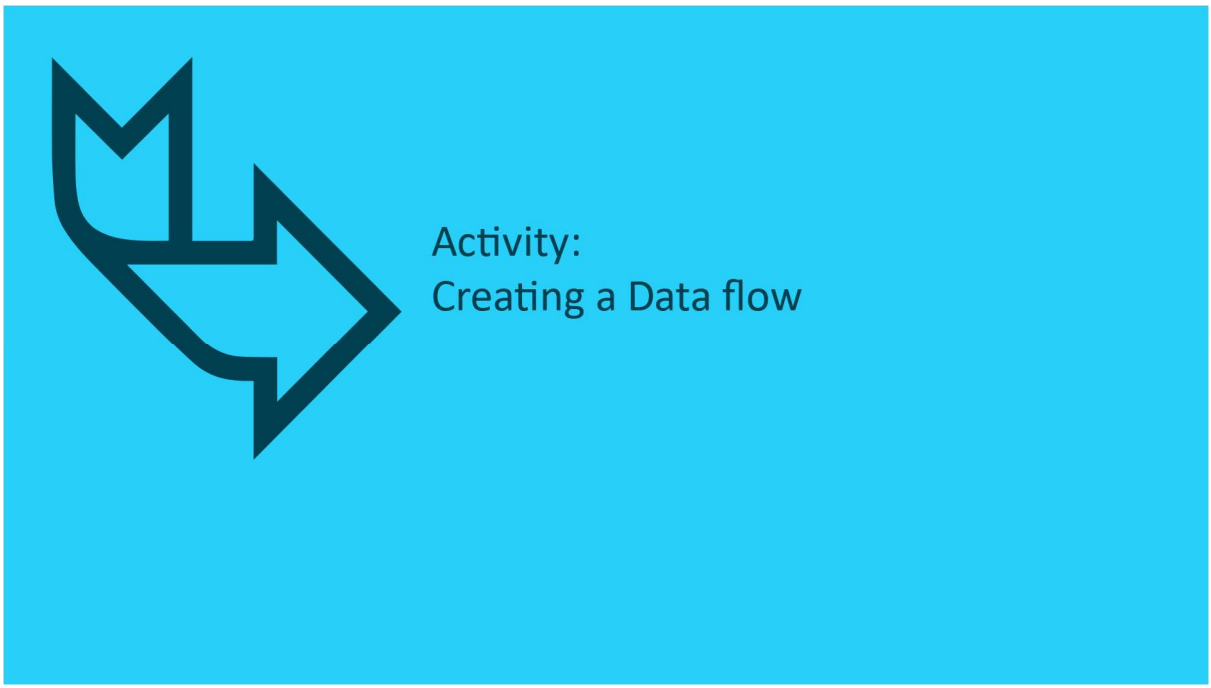
- a dataset object (defined separately)
- an inline dataset (defined in the sink)

Most data formats are only supported by one type of sink

A Cache Sink is stored in the Spark cache and can be reused by lookups

Data Preview shows that data as it would appear, but does not write to the destination

87



We are going to add a Data Flow to the Copy operation you previously created. It takes the data from the ***load.Targets*** table and apply transformations before loading into the ***stage.Targets*** table

1. On the Pipeline that loaded the data from the file into the SQL Database, add a Data Flow to the Pipeline.
2. In the Data Flow Properties, Settings tab, add a new Data Flow.
3. Click on Data Flow Debug, to start the cluster for debugging, accept the default settings. This will take some time to start.
4. Click on the Source area to add a Source.
5. In the Source Settings:
    - Source type: *Dataset*
    - Dataset: select *DestinationDataSet* from the previous copy operation

Click on Projection Tab, If there are no columns listed, click on *Import projection.* The column names and data types should be listed that match the data in the table.

Click on Data Preview tab. Click Refresh to view the data in the source.

Add an Unpivot Transformation by clicking on the small + at the right corner of the Source symbol in the data flow. Select Unpivot from the list of transformations.

In the Unpivot settings, Ungroup by, select *Year* and *EmployeeID*. You might need to use the Expression Builder, both Column name and Expression will just be the name of the column.

In the Unpivot Key, type *Month* as the column name and string as data type.

In the Unpivoted columns, set

- Column Arrangement as Normal
- Column Name as Target
- Column Type as String

Check the Data Preview. There should be 4 columns: Year, EmployeeID, Month and Target.

Add a Filter Transformation to remove the rows with "-" in the Target column.

Use the expression builder to add ***Target != "-"***

Check the Data Preview to ensure the rows have been removed.

Add a Derived Column Transformation to combine the Year and Month to create a date column. Assume the day as the 1st of the month. We need to remove the 'M' from the Month column. Also multiply the Target value by 1000.

- TargetDate = toDate(Year +'-' + replace(Month, 'M')+ '-01')
- Target = toDecimal(Target) * 1000

Add a Select Transformation. Remove the Year and Month columns which are no longer needed.

Add a sink using the Azure SQL Linked Service and creating a table stage.Targets

Save the Data Flow and the Pipeline. Before running the Pipeline, delete *Load.Targets* and *Stage.Targets* using the SQL Management Studio (this is a process we will need to automate in the future).

On the Pipeline tab in ADF, click on the background of the pipeline and click *Debug* to run the complete paipline.

Transforming data with SQL Stored Procedures

For transformations within an Azure SQL Database environment it is usually more efficient to use SQL Stored Procedures

- The source and destination tables must be in the same Azure SQL database
- Write a stored procedure in T -SQL and store in the Azure SQL Database
- Use the SQL Server Stored Procedure activity in ADF to call the procedure

An Alternative method is to use ADF Copy activity

- When configuring the SQL Sink specify a stored procedure and a user defined Table Type
- This enable data to be retrieved from one database and the stored procedure used to transform the data as it is being written to another database.

89

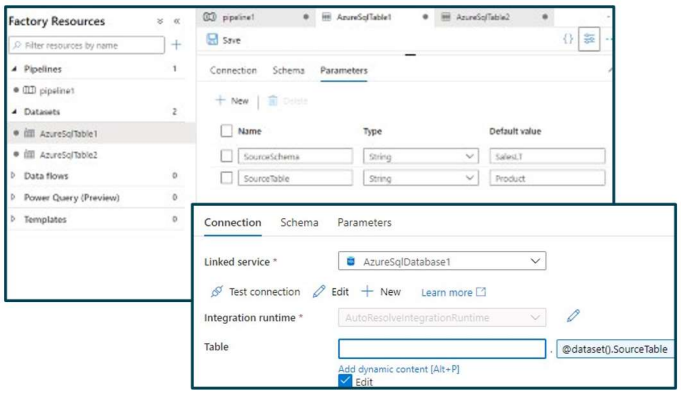
In this activity, you will copy the Resellers table from the source database to the load schema. Then use a Stored procedure to correct a problem with values in the BusinessType column.

6. Create a new Pipeline called Resellers.
7. Create a Linked Service to connect to the Adventureworks2020 Source database (Check with your instructor for connection details).
8. Add a copy activity to the Pipeline to copy from the Adventureworks2020.dbo.Resellers to ossqldbxxx.load.Resellers.
9. Run this activity in Debug Mode. Check that the data is copied to your table
10. Create a Stored Procedure in ossqldbxxx.stage.uspResellers. This should SELECT from load.Resllers and insert into stage.Resellers. Use a CASE expression to correct the [BusinessType] column 'Ware House' with 'Warehouse'.
11. Execute the Stored Procedure, and check the data is copied as expected.
12. DROP the stage.Resllers and load.Resellers tables.
13. Add a Stored Procedure activity to the Pipeline in Data Factory, and configure it to execute your stored procedure.
14. Run the whole pipeline and check the 2 tables are populated as expected.
15. In the Copy activity, Sink properties, add a DROP statement in the Precopy option. It should check for the existence of the table.
    - IF EXISTS (SELECT OBJECT_ID('load.Resellers'))
      DROP load.Resellers
16. Add similar code to the beginning of your stored Procedure to check for the existence of the stage.Resellers.
17. Run the Pipeline again to check it drops and creates the tables correctly.

ADF Parameters

Azure Data Factory Parameters can be created on a Pipeline and used to change the behaviour of activities, linked services and datasets.

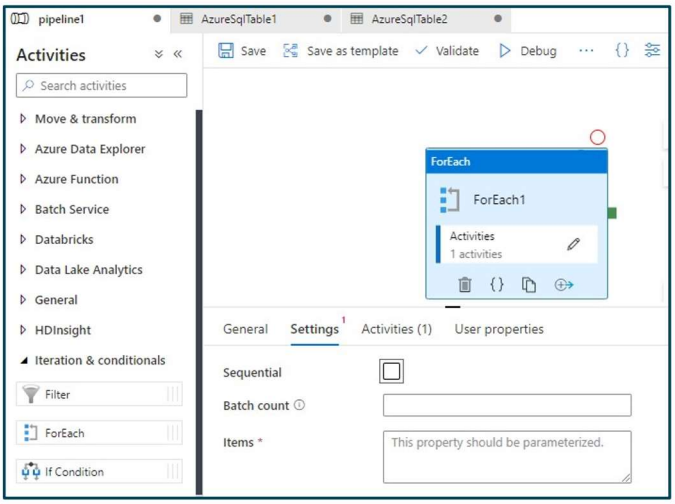
- Define the parameter names, data types and default values
- The Default values are place holders and will be replaced
- Use the Edit option and Add dynamic content
- Use the expression builder to define the value
- In the parent activity, define the parameters to pass in



Repeated activities

With values exposed as Parameters, an activity can be repeated by changing parameter values.

e.g. For Each – repeat an activity based on a parameter containing an array of values



In this activity, you will create parameters for the data source, sink and stored procedure so that by changing the parameter, a different table will be copied.

Summary

Use Azure Data Factory to orchestrate data movements in Azure

Create Pipelines for each different type of operation

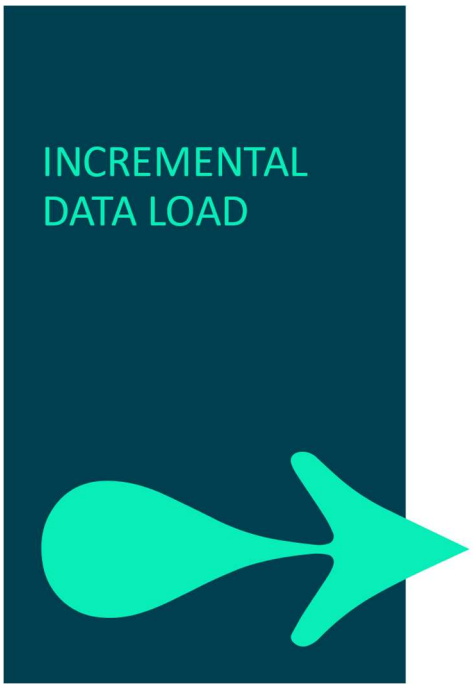
Use Transforming Data flow when complex operations are needed

Use SQL Stored procedures when source and sink are in the same SQL database

Parameterise pipeline to reuse

94

Chapter 7 – Incremental Data Load



Identifying changed data
Extracting changed data
Loading changed data

96

Incremental Load Process

Identify the data that has changed since the last load

Extract the data that has been changed

Load the data into the Data Warehouse

- Does the data need to be added as new data?
- Should existing data be changed?

Record the end of the changed data, ready for the next load cycle

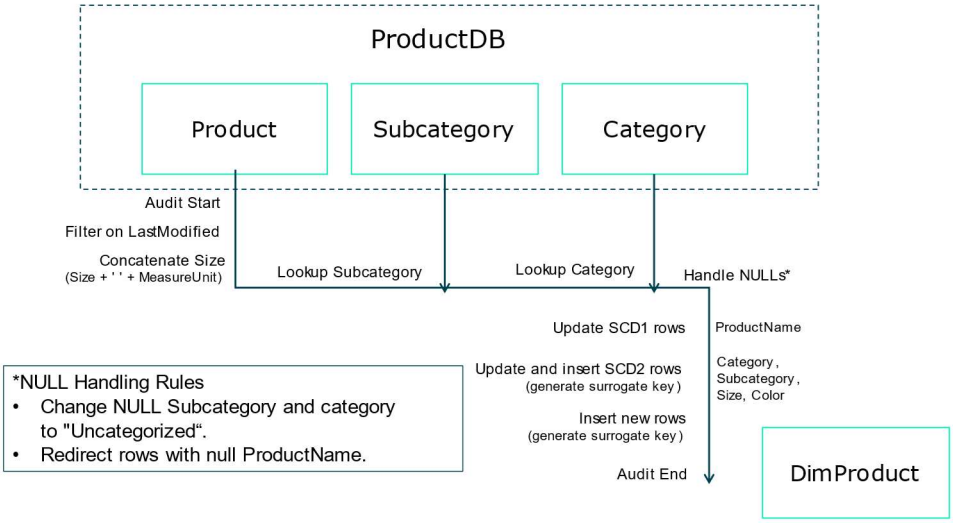
97

Planning Extraction Windows

- How frequently is new data generated in the source systems, and for how long is it retained?
- What latency between changes in source system and reporting is tolerable?
- How long does data extraction take?
- During what time periods are source systems least heavily used?

98

Documenting Data Flows



*NULL Handling Rules
- Change NULL Subcategory and category to "Uncategorized".
- Redirect rows with null ProductName.

99

Slowly Changing Dimensions

- Types of change to a dimension member:
  - Type 1: Changing attributes are updated in the dimension record

Key	AltKey	Name	Phone	City
101	C123	Mary	5551234	New York

Key	AltKey	Name	Phone	City
101	C123	Mary	5554321	New York

  - Type 2: Historical attribute changes result in a new record

Key	AltKey	Name	Phone	City	Current
101	C123	Mary	5551234	New York	True

Key	AltKey	Name	Phone	City	Current
101	C123	Mary	5551234	New York	False
102	C123	Mary	5551234	Seattle	True

  - Type 3: The original and current values of historical attributes are stored in the dimension record

Key	AltKey	Name	Phone	OriginalCity	CurrentCity	EffectiveDate
101	C123	Mary	5551234	New York	New York	1/1/00

Key	AltKey	Name	Phone	OriginalCity	CurrentCity	EffectiveDate
101	C123	Mary	5551234	New York	**Seattle**	**6/7/11**

100

Options for Extracting Modified Data

Extract all records

Store a primary key and checksum

Use a datetime column as a “high water mark”

Use Change Data Capture

Use Change Tracking

Use Temporal Table s

n.b. Change Data Capture, Change Tracking and Temporal tables require the source database to support these features.

→ CDC is configured with Stored procedures, and copies data as it is changed to system tables

→ Change Tracking uses a version no. to identify that a row has changed.

→ Temporal Tables used System- versioned tables

101

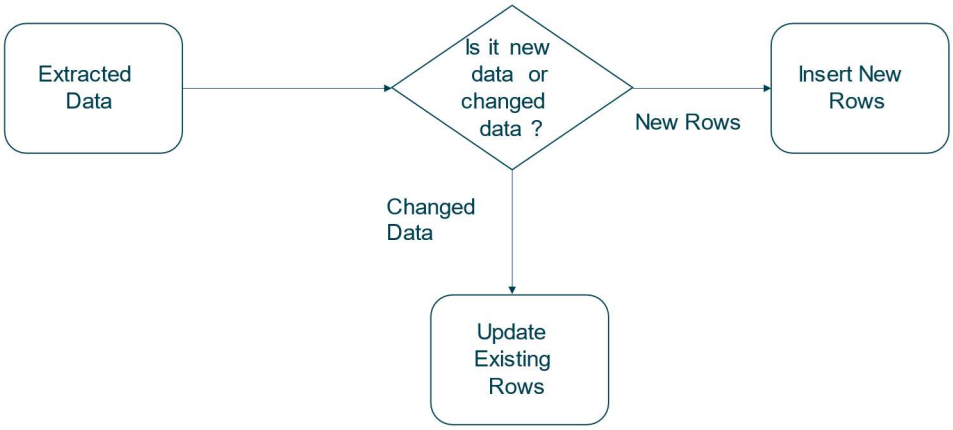
Extracting Rows Based on a Datetime Column

1. Note the current time
2. Retrieve the last extraction time from an extraction log
3. Extract and transfer records that were modified between the last extraction and the current time
4. Replace the stored last extraction value with the current time

102

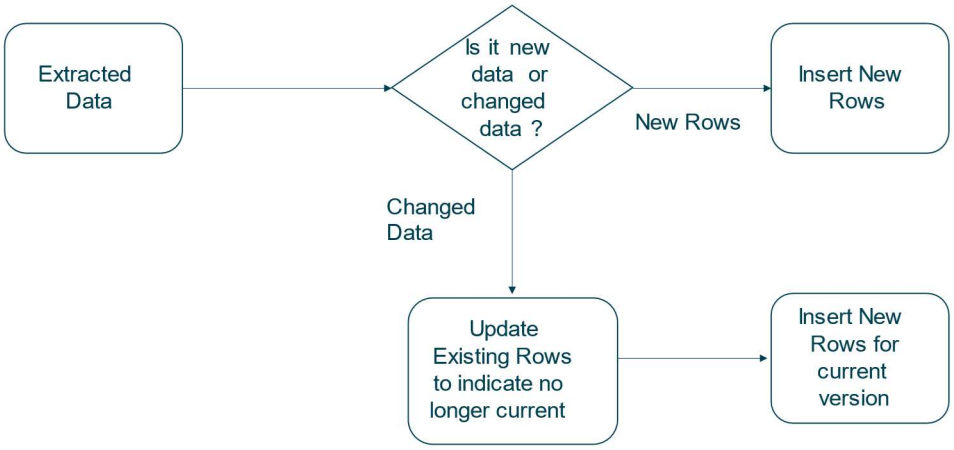
In this activity, you will copy the Products and archived products, and then combine them into a single table.

Loading Data



104

Slowly Changing Dimension



105

Activity:
Creating pipelines for changing data

Identify changing data in source systems

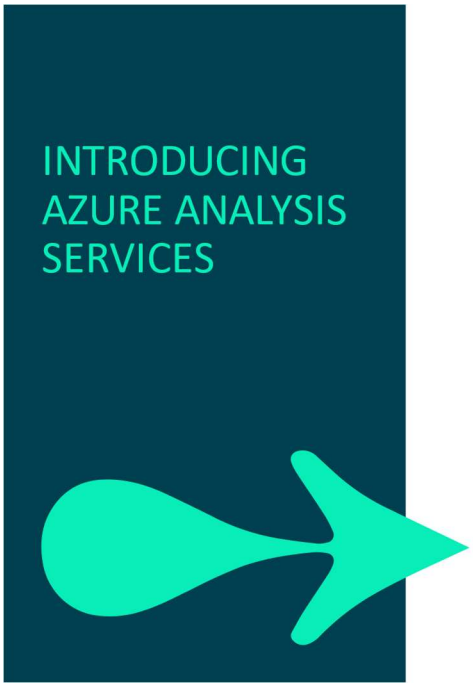
Identify change capture requirements

Data Lake storage can be used to store all changes, incase needed for the future

Summary

107

Chapter 8 – Introducing Azure Analysis Services



Introduction to Tabular Data Models in SQL Server Analysis Services

Creating a Tabular Data Model

Using a SQL Server Analysis Services Tabular Model in an Enterprise BI Solution

109

Creating an Azure Analysis Server

Create an Azure Analysis Server:

- Server name must be globally unique
- The Administrator is an Azure AD Account or group
- Pricing is based Query Processing Units
- A Backup Storage location is usually defined on a storage account

A Server can be paused and resumed as required.

Models can only be viewed when the server is running

111

Options for Creating a Tabular Data Model Project in SQL Server Analysis Services

Create a tabular data model project by using:

- SQL Server Data Tools for BI templates:
  - Analysis Services Tabular Project
  - Import from PowerPivot
  - Import from Server (Tabular)
- SQL Server Management Studio:
  - Restore from PowerPivot
- Key features in Analysis Services tabular databases
  - DirectQuery mode
  - Row -level and object -level security
  - Partitions

112

Using SQL Server Data Tools to Develop a Tabular Data Model in SQL Server Analysis Services

- Single development environment for BI developers that supports a range of features:
  - Templates to create projects
  - Integrated project life cycle management and source control
  - Debugging tools
  - Building and deployment directly to test and production servers
- Tabular Model Designer:
  - Data view and diagram view
  - Measure grid
- Get Data

113

The Workspace Database

- Created on SSAS instance when tabular data model project is created
- Contains all data and metadata of project
- Can use local or remote SSAS instance
- Configure the workspace database:
  - Workspace Server
  - Workspace Retention

Importing Tables

- Create data source connections for a wide range of connection options including common third -party databases
- Filter out columns that are not required for analysis:
  - Improves performance
  - Simplifies user experience
- Provide table aliases for ease of use

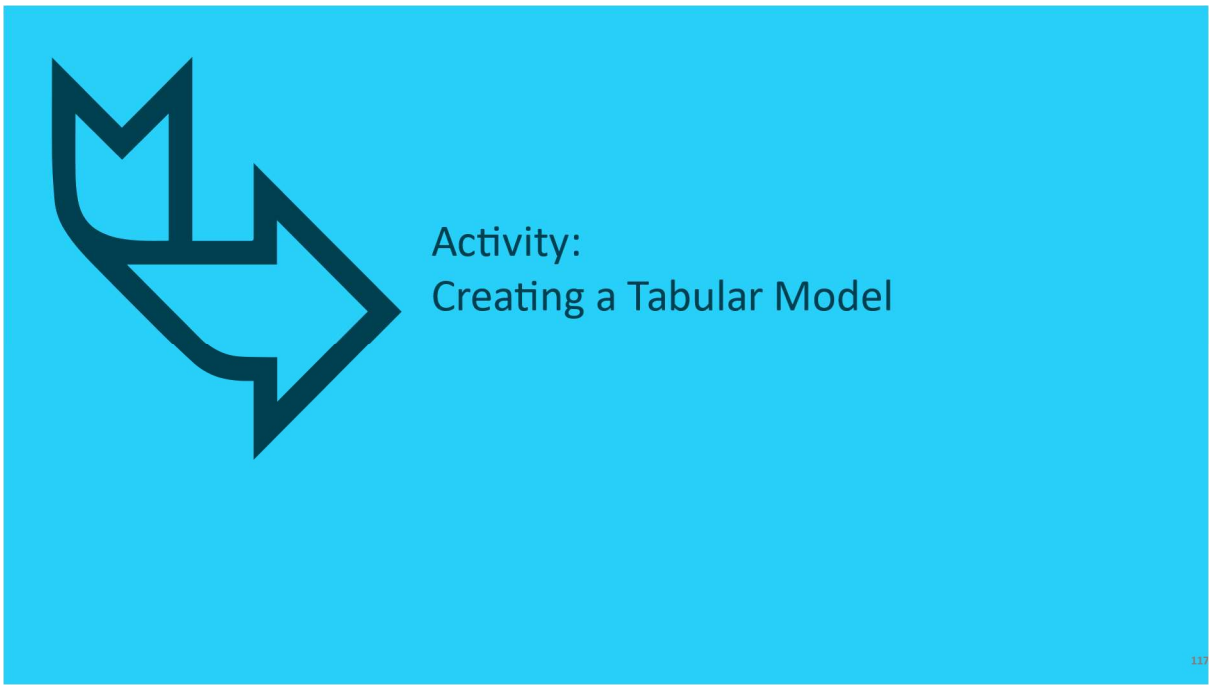
115

Defining Measures

- Measures are numeric aggregations of business metrics
- They are usually based on a simple DAX function that aggregates a numeric column value

```
Revenue:=Sum([SalesAmount])
```

116



In this activity you will create an Analysis Services Server and load data from your dw schema.

1. In your storage account, create a container called osasbackup
2. Create an Azure Analysis Server:
    a. Select the D1 pricing tier
    b. Select your Azure AD account as administrator.
    c. Use your storage account and osasbackup container for the backup location
3. Run Visual Studio 2019. Create a New Project of type Analysis Services Tabular Model. Use an appropriate name and folder location for the project
4. In the Tablular Model Designer dialog, select ***Integrated Workspace*** and select ***SQL Server 2019/Azure Analysis Services (1500)***
5. On the *Extensions* Menu, click *Model* and select *Import from Data Source*. Iport all of the tables and all of the columns, since we have already cleansed the data.

Managing Relationships

- Automatically recognize relationships based on foreign keys
- Manually create relationships when they are not explicitly defined
- Only one relationship between two tables can be active
- To define role -playing dimensions, import the same table multiple times

118

Configuring Columns

- Specify column data types
- Create calculated columns based on DAX expressions
- Specify sort orders for columns
- Hide and rename columns

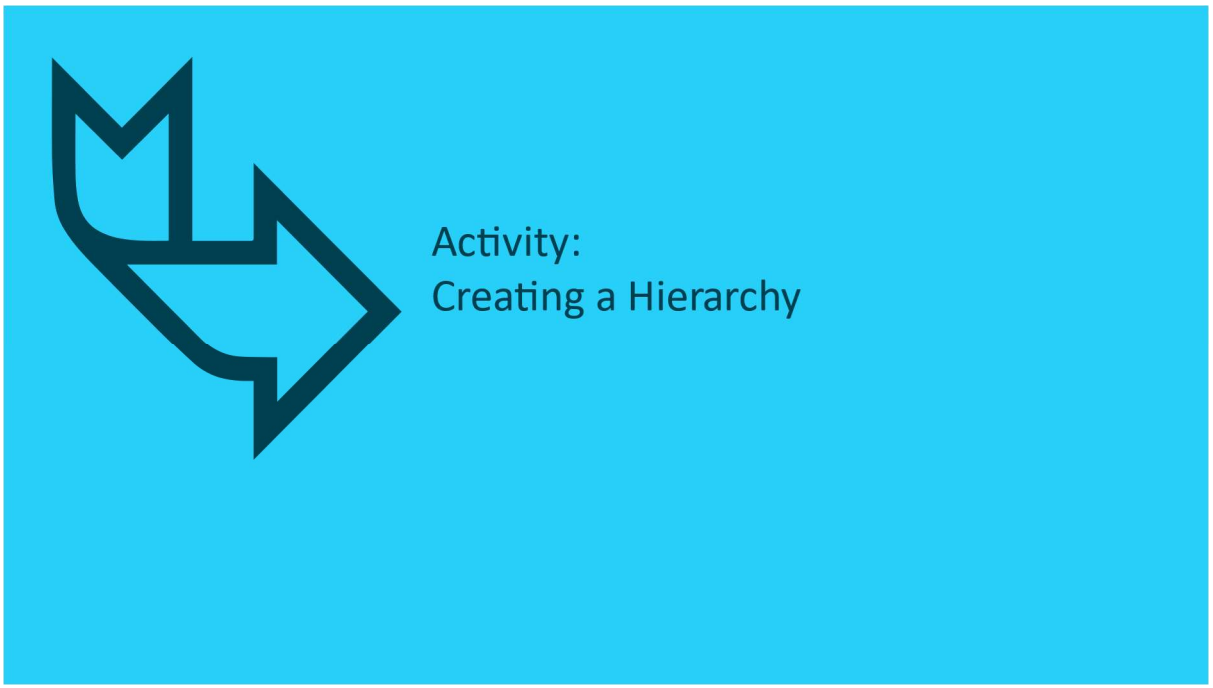
119

In this Activity you will create the relationships in the data model. Most are imported, but they need to check and adjusted.

Creating Hierarchies

- Hierarchies enable drill -up and drill -down aggregations
- Common examples:
  - Product Category, Product Subcategory, and Product Name
  - Year, Quarter, Month, and Date
- Create hierarchies in the diagram view
- Ragged hierarchies

121



In this practice you will create the Hierarchies that were identified in Chapter 2.

DimDate -> Calendar Date and Financial Date

DimGeography -> locations

DimProduct -> Products

DimSalesTerritory -> Regions.

Perspectives

- Provide a simplified view of complex data models
- Select tables, measures, and columns to include in perspective
- Not a method of implementing security

123

Partitions

- Divide tables into logical partitions
- Separate frequently changing data from static data
- Create partitions in workspace and deployed databases

124

Analysis Services Processing Types

- Processing reads data from the data source and populates the model
  usually drops the data before reloading
- Process Types:
  Process Default – processes all changed objects and related objects
  Process Full – processes all objects
  Process Data – does not change the object, just reloads the data
  Process Clear – removes all data leaving the objects
  Process Add – Adds and processes additional data
- Processing databases, tables, and partitions

125

More information on partition processing

https://docs.microsoft.com/en-us/analysis-services/tabular-models/process-database-table-or-partition-analysis-services

DirectQuery Mode

- Bypass xVelocity in-memory storage
- Retrieve data directly from a relational database
- Take advantage of powerful server hardware
- Appropriate for very large data sets
- Some feature limitations to consider

126

More information on Direct Query Mode

https://docs.microsoft.com/en-us/analysis-services/tabular-models/directquery-mode-ssas-tabular

Security for Tabular Data Models

- Create roles to group users according to security requirements
- Apply database -level permissions to roles:
  - Read
  - Read and Process
  - Process
  - Administrator
  - None
- Use DAX to create filters that define row -level security
- Use JSON -based metadata, TMSL, or TOM to configure table -level and column -level security

127

More information on security

https://docs.microsoft.com/en-us/analysis-services/tabular-models/roles-ssas-tabular

Automating processing of Analysis Services Model

Use Azure Automation Account

- Install SQLServer PowerShell modules
- Create a Service Principal with server administrator permissions
- Create a run book with the sample Refresh -Model.ps1 script
- Create a schedule to run the RunBook

To Automate with Data Factory

- Add a Webhook to the RunBook
- Add a web activity to Azure Data Factory pipeline to trigger the Webhook

128

More information on automating refresh:

https://docs.microsoft.com/en-us/azure/analysis-services/analysis-services-refresh-azure-automation

Backup an Analysis Services Database

Configure an Azure Storage account container as a backup location in the portal

- Manual backup
  Use the SSMS to backup a database

- Automated backup
  UsePowerShell *Backup* *-ASDatabase* cmdlet in an Azure RunBook

Backup and Restore can be used to move a database from an on -premise AS server to Azure AS. Backup file must be copied to the storage account.

129

More information on backups:

https://docs.microsoft.com/en-us/azure/analysis-services/analysis-services-backup

Create an Azure Analysis Server

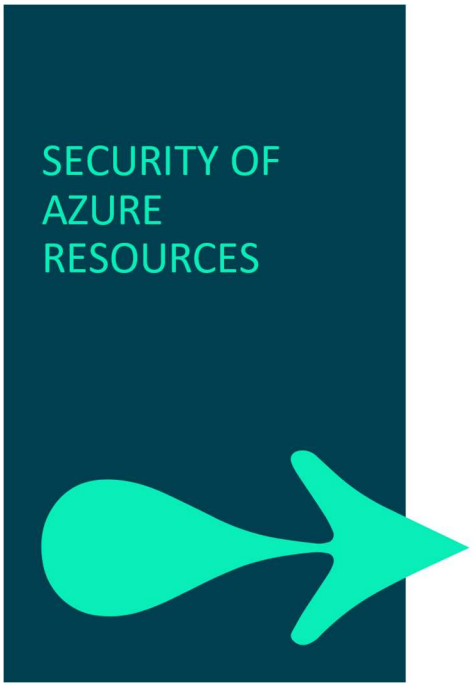
Use Visual Studio with Data Tools to create AS Tabular models

Automate the refresh of data as part of the D ata Warehouse load process

Pause Server when not needed to reduce costs

130

Chapter 9 – Security of Azure Resources



Overview of security in Azure
Network Security
Identity and access management
Encryption capabilities
Securing Storage Accounts
Securing Azure SQL Databases
Securing Azure Analysis Services

132

Overview of Security in Azure

Network security

Identity and access management

Encryption capabilities built into Azure

133

Network Security

Securing your network from attacks and unauthorized access is an important part of any architecture

Internet protection	Firewalls	DDoS protection	Network security groups
Assess the resources that are internet-facing, and to only allow inbound and outbound communication where necessary. Make sure you identify all resources that are allowing inbound network traffic of any type	To provide inbound protection at the perimeter, there are several choices: • Azure Firewall • Azure Application Gateway • Azure Storage Firewall	The Azure DDoS Protection service protects your Azure applications by scrubbing traffic at the Azure network edge before it can impact your service's availability	Network Security Groups allow you to filter network traffic to and from Azure resources in an Azure virtual network. An NSG can contain multiple inbound and outbound security rules

134

Identity and access

Authentication

This is the process of establishing the identity of a person or service looking to access a resource. Azure Active Directory is a cloud - based identity service that provide this capability

Authorization

This is the process of establishing what level of access an authenticated person or service has. It specifies what data they're allowed to access and what they can do with it. Azure Active Directory also provides this capability

Azure Active Directory features

Single sign-on

Enables users to remember only one ID and one password to access multiple applications

Apps & device management

You can manage your cloud and on -premises apps and devices and the access to your organisation's resources

Identity services

Manage Business to business (B2B) identity services and Business - to -Customer (B2C) identity services

Encryption

Encryption at rest

Data at rest is the data that has been stored on a physical medium. This could be data stored on the disk of a server, data stored in a database, or data stored in a storage account

Encryption in transit

Data in transit is the data actively moving from one location to another, such as across the internet or through a private network. Secure transfer can be handled by several different layers

Encryption on Azure

Raw encryption

Enables the encryption of:

- Azure Storage
- V.M. Disks
- Disk Encryption

Database encryption

Enables the encryption of databases using:

- Transparent Data Encryption

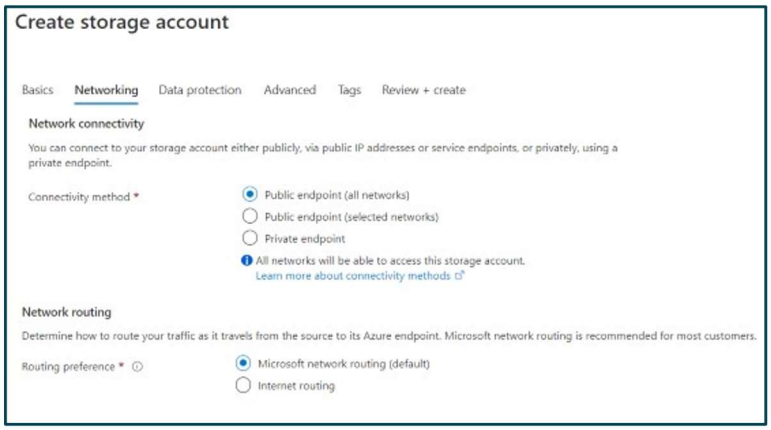
Encrypting secrets

Azure Key Vault is a centralized cloud service for storing your application secrets

136

Securing Storage Accounts - Network

- Use a virtual network and ensure the storage account is only accessible from the virtual network
- Allow access from trusted Microsoft services e.g. Data Factory
- Add specific IP address ranges when uploading from on -premise locations



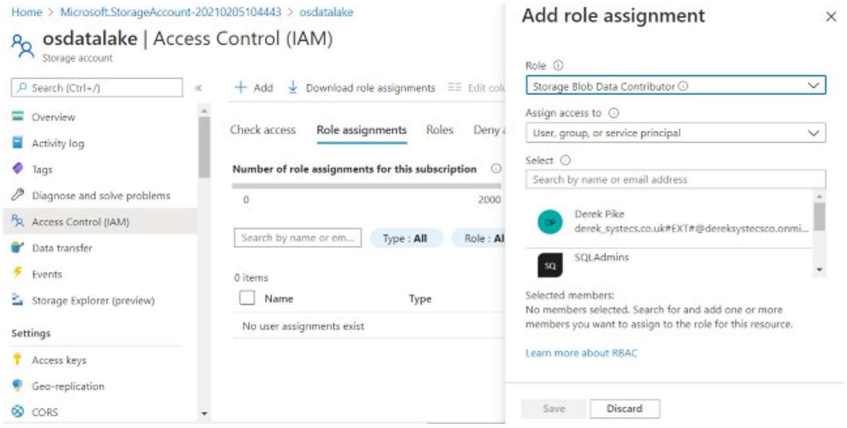
137

Securing Storage Accounts – Identity and Access

Use Role based Access Control (RBAC) when possible

Create Service Identities if necessary

Only users and groups displayed initially. Start to type an identity to search



138

Storage Accounts – Shared Access Signatures

A Shared Access Signature (SAS) can be used to give limited access to a container, folder or file.

Can be limited by:

- Service
- Resource Type
- Permissions
- Time Period
- IP Address
- Protocol

139

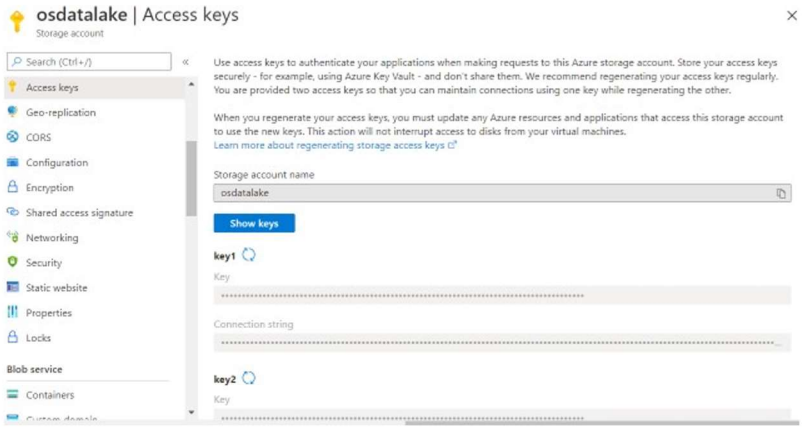
Storage Accounts – Access Key

Access keys give full access to a storage account

Use only if other methods not available.

Use the Azure Key Vault to store securely

2 keys provided to ease regular regeneration



140

Azure SQL Database – Network Security

There are a number of ways you can control access to your Azure SQL Database or Data Warehouse over the network

Server-level firewall rules	Database level firewall rules
These rules enable clients to access your **entire Azure SQL server** , that is, all the databases within the same logical server Can be configured from the Azure Portal	These rules allow access to an individual database on a logical server and are stored in the database itself. For database -level rules, only **IP address rules** can be configured Can only be configured using SQL

141

Azure SQL Database – Identity and Access

Authentication	Authorization
SQL Database supports two types of authentication: SQL authentication and Azure Active Directory authentication Set AAD SQL Admin account in the Azure Portal All other accounts are created using SQL	Authorization is controlled by permissions granted directly to the user account and/or database role memberships. A database role is used to group permissions together to ease administration Configured using SQL

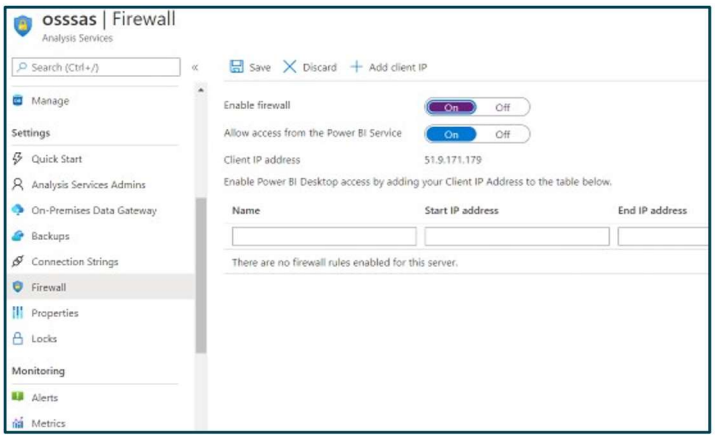
142

Azure Analysis Services - Firewall

Enable the Firewall to restrict network access to the server.

Create rules allowing access from clients to manage, deploy and use databases

Enable access from the Power BI service to use the databases.

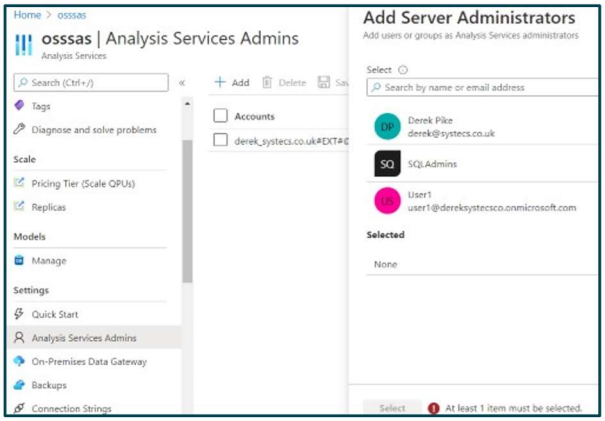


143

Azure Analysis Services – Identity and Access

AS can only accept AAD accounts and groups as administrators

Other access is configured by creating roles using SSMS



144

Summary

- Create Virtual Networks
- Configure all resources to be accessible on the Virtual Network
- Allow access to Azure Resources e.g. ADF and Power BI
- Block access from Public IP addresses
- Create rules to allow limited access when needed
- Create AAD identities
- Use Role Based Access Control (RBAC) when possible
- Limit use of Storage Account Access Keys

145

Chapter 10 - Management of Azure Resources

Automation of operations
Monitoring services
Data Backup and recovery

147

Automating operations

There are a number of ways to automate operations depending on the service:

Azure Data Factory pipeline:

- runs can be executed manually, using a REST API or PowerShell
- Pipeline triggers

Azure SQL operations:

- Automation Runbook
- Elastic Job

Azure Analysis Services

- Automation Runbook

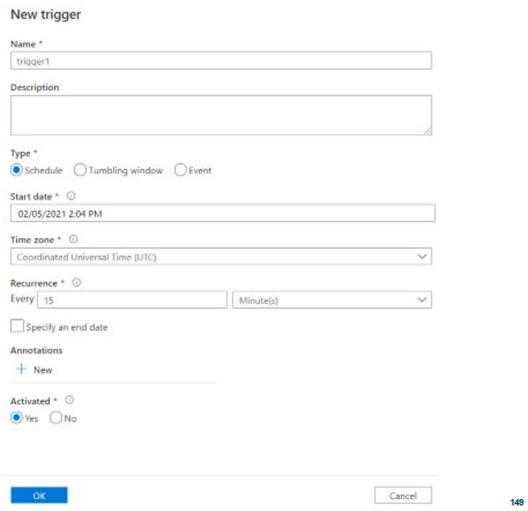
148

ADF – Pipeline Triggers

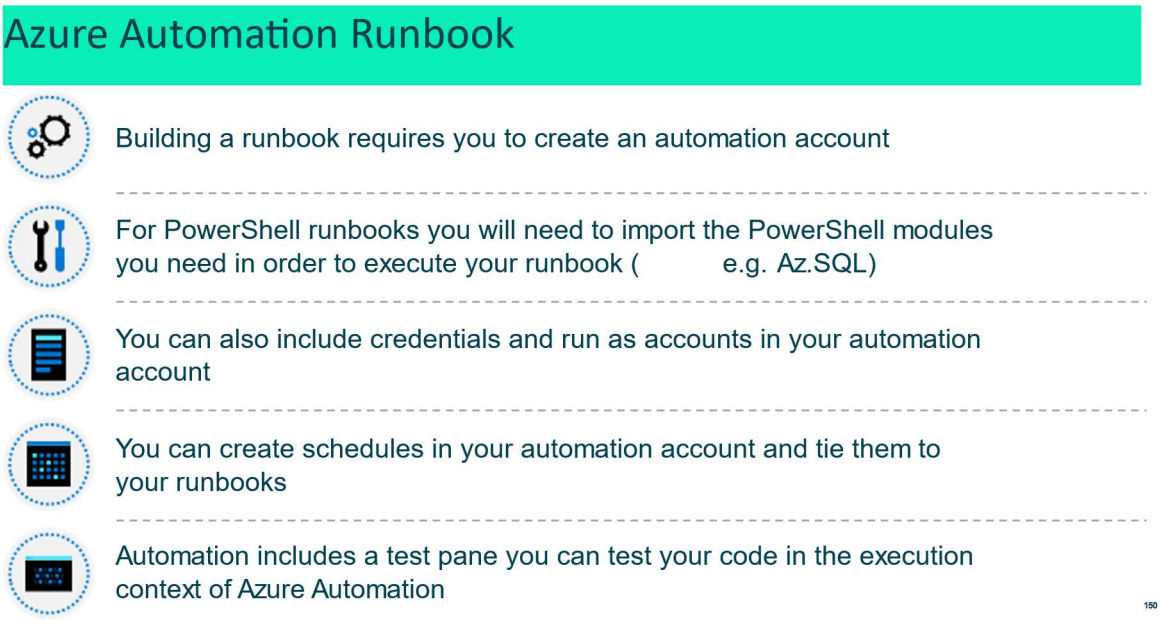
Pipeline Triggers are created as part of the Pipeline

Triggers can be:

- Schedule – at a particular time, or repeating
- Tumbling Window – reoccurring from a fixed time
- Event – blobs being created or deleted from a Storage container



https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers

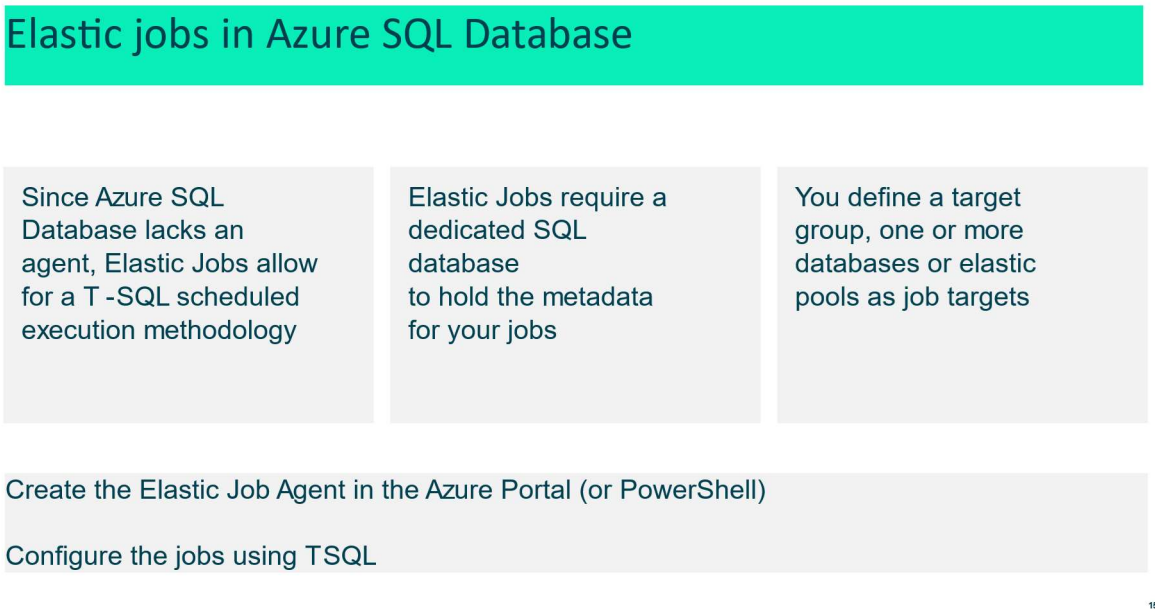


Creating an Automation Account

https://docs.microsoft.com/en-us/azure/automation/automation-quickstart-create-account

Creating a Runbook

https://docs.microsoft.com/en-us/azure/automation/automation-quickstart-create-runbook



Overview of Elastic Jobs

https://docs.microsoft.com/en-us/azure/azure-sql/database/elastic-jobs-overview

Creating and managing an Elastic Job using T-SQL

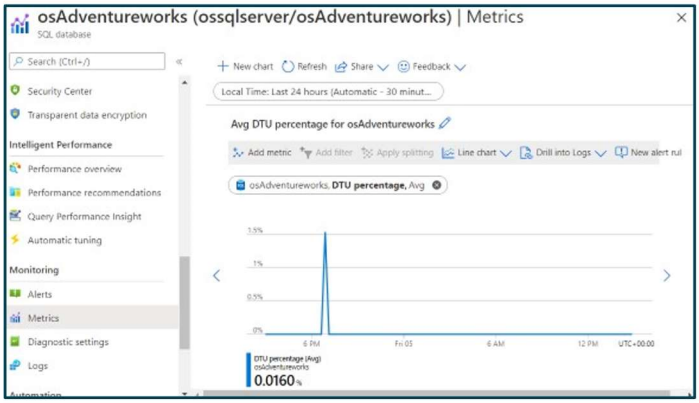
https://docs.microsoft.com/en-us/azure/azure-sql/database/elastic-jobs-tsql-create-manage

Monitoring Azure Services

The Azure Monitor can be used to monitor the state of any Azure Service

Most commonly used metrics are display in the Overview tab of the resource

All metrics for a resource can be viewed.



152

Azure Monitor Alerts

Alerts can be configured based on the Metrics for a resource

- Choose the Resource
- Create the condition with a threshold for the metric e.g. “DTU % rises above 70%”
- Specify an Action.
  - May include a notification – email, SMS
  - May include an action – execute an Automation Run Book
- Name the Alert Rule and enable

153

Azure Storage Backup

Storage accounts are automatically copied to multiple storage systems in an Azure Region.

- Locally -redundant storage (LRA) - 3 copies in a region
- Geo -redundant storage (GRS) – 6 copies: 3 each in 2 different regions
- Read -access Geo -redundant storage (RA -GRS) – as above but always accessible

Storage accounts can also have the following configurations

- Point -in -time restore for containers
- Soft delete for blobs
- Versioning for blobs

154

https://azure.microsoft.com/en-gb/blog/microsoft-azure-block-blob-storage-backup/

https://docs.microsoft.com/en-gb/azure/storage/blobs/point-in-time-restore-overview

https://docs.microsoft.com/en-gb/azure/storage/blobs/soft-delete-blob-overview

https://docs.microsoft.com/en-gb/azure/storage/blobs/versioning-overview

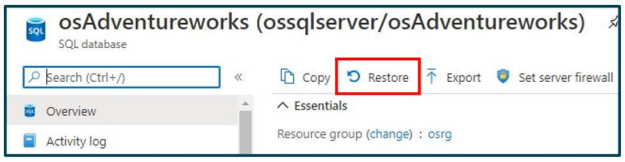
SQL Database Backup and Restore

Azure SQL Database is automatically backed up

- Full backup every week
- Differential Backup every 12 – 24 hours
- Transaction log backup every 5 – 10 minutes
- Backup Retention can be configured from the logical server (Default is 7 – 35 days)
- Longer term backup is also available

Restore a Point -in -time backup to a new database from the portal

Objects can be copied from the restored database to the original



155

Overview of Automated backups

https://docs.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?tabs=single-database

Changing the backup retention period

https://docs.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?tabs=single-database#change-the-pitr-backup-retention-period

Database Recovery

https://docs.microsoft.com/en-us/azure/azure-sql/database/recovery-using-backups

Azure Data Factory

All objects in the Azure Data Factory are documented as json scripts

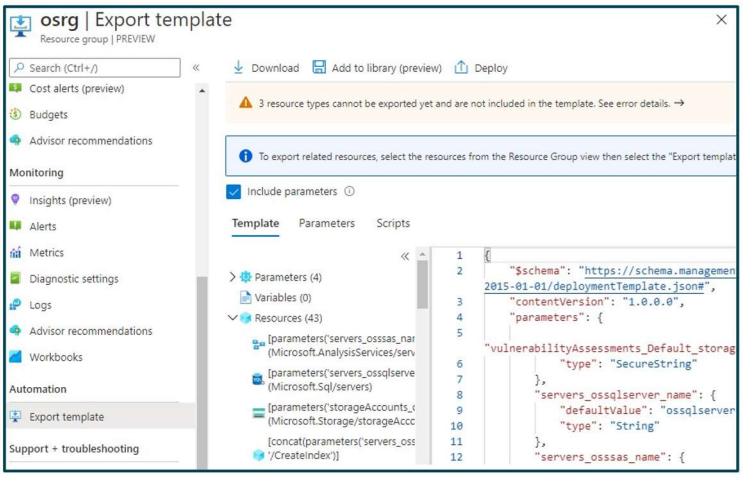
Connecting ADF to a repository, e.g. github or Azure Devops , ensures code is stored securely

156

Azure Templates

All Azure services in a resource group can be documented using Templates.

Templates make it easy to re create objects when required, or move from development to production environment



157

Overview of working with Azure Templates

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview#

Disaster Recovery

Make a plan!

- Ensure backup of data is documented
  - What automatic backups are in place?
  - What has to be manually configured?
- Ensure recovery procedures are documented
  - How is a particular resource recovered ?

158