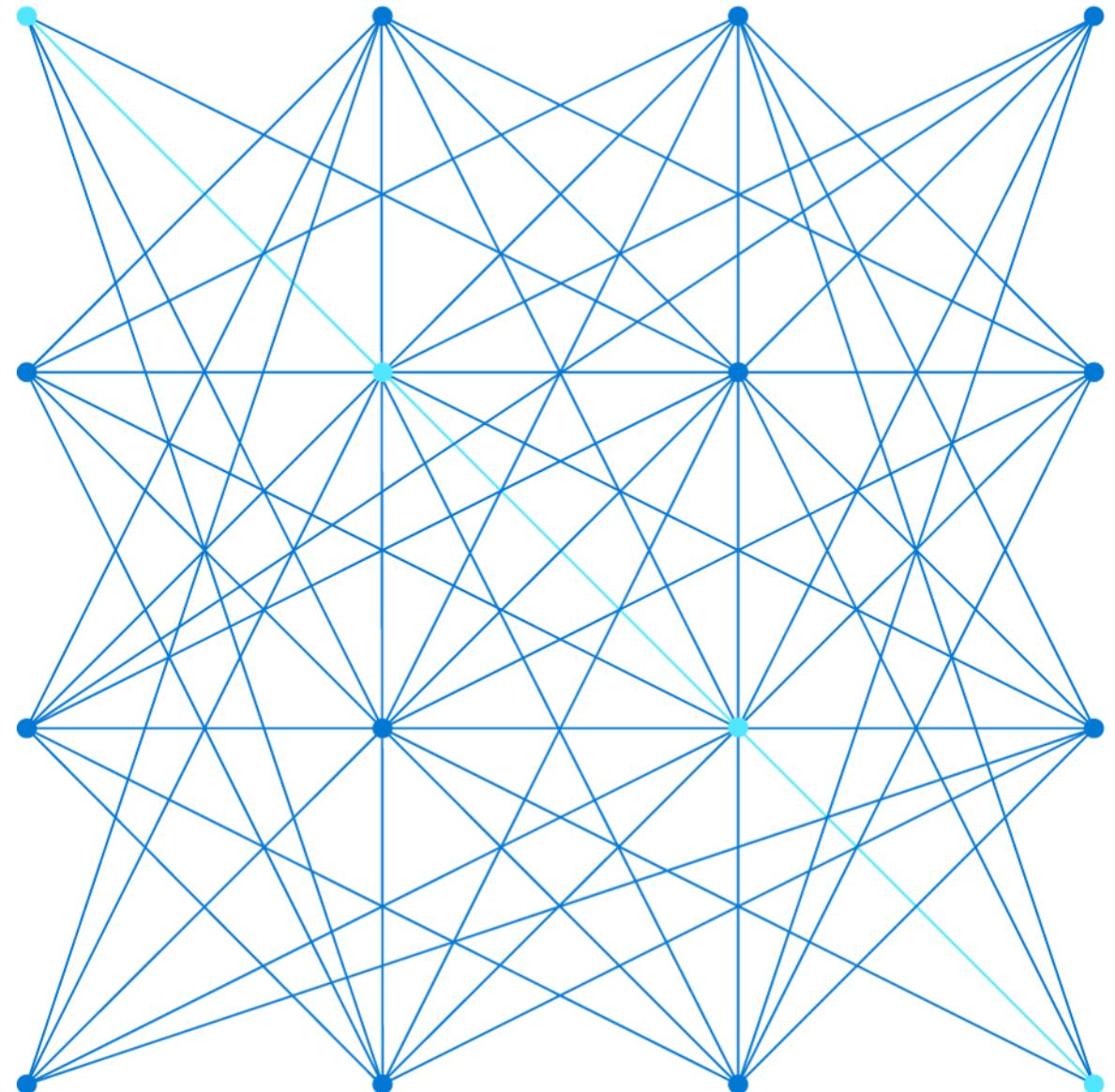


# Implementing an Azure Data Solution [DP-200]



# Tissana Tanaklang

Software and Solution Development Trainer  
Iverson Training Center Co., Ltd.  
tissana@iverson.co.th , tissana\_t@hotmail.com



- Master of Science Program in Software Engineering King Mongkut's University of Technology Thonburi
- Bachelor of Science Program in Computer Science Naresuan University
- Microsoft Certified Trainer (MCT)
- Microsoft Certified Azure Data Engineer Associate
- Microsoft Certified Solutions Associate - Web Application Development
- Microsoft Certified Azure Fundamentals
- Microsoft Certified Azure Data Fundamentals
- Microsoft Certified Azure AI Fundamentals



# Agenda



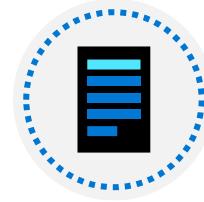
About this course



Course agenda



Audience



Prerequisites

# About this course

In this course, the students will implement various data platform technologies into solutions that are in line with business and technical requirements including on-premises, cloud, and hybrid data scenarios incorporating both relational and No-SQL data. They will also learn how to process data using a range of technologies and languages for both streaming and batch data

The students will also explore how to implement data security including authentication, authorization, data policies and standards. They will also define and implement data solution monitoring for both the data storage and data processing activities. Finally, they will manage and troubleshoot Azure data solutions which includes the optimization and disaster recovery of big data, batch processing and streaming data solutions

# Course agenda

## Module 1

### Azure for the Data Engineer

**Lesson 01** – Explain the evolving world of data

**Lesson 02** – Survey the services in the Azure Data Platform

**Lesson 03** – Identify the tasks that are performed by a Data Engineer

**Lesson 04** – Describe the use cases for the cloud in a case study

## Module 2

### Working with Data Storage

**Lesson 01** – Choose a data storage approach in Azure

**Lesson 02** – Create an Azure Storage Account

**Lesson 03** – Explain Azure Data Lake Storage

**Lesson 04** – Upload data into Azure Data Lake

# Course agenda (*continued #1*)

## Module 3

Enabling team based Data Science with Azure Databricks

**Lesson 01** – Explain Azure Databricks

**Lesson 02** – Work with Azure Databricks

**Lesson 03** – Read data with Azure Databricks

**Lesson 04** – Perform transformations with Azure Databricks

## Module 4

Building globally distributed databases with Cosmos DB

**Lesson 01** – Create an Azure Cosmos DB database built to scale

**Lesson 02** – Insert and query data in your Azure Cosmos DB database

**Lesson 03** – Build a .NET Core app for Azure Cosmos DB in Visual Studio Code

**Lesson 04** – Distribute your data globally with Azure Cosmos DB

# Course agenda (*continued #2*)

## Module 5

Working with relational data stores  
in the cloud

**Lesson 01** – Explain SQL Database

**Lesson 02** – Explain SQL Data Warehouse

**Lesson 03** – Provision and load data in Azure SQL  
Data Warehouse

**Lesson 04** – Import data into Azure SQL Data  
Warehouse using PolyBase

## Module 6

Performing real-time analytics with Stream  
Analytics

**Lesson 01** – Explain data streams and event  
processing

**Lesson 02** – Data ingestion with Event Hubs

**Lesson 03** – Processing data with Stream  
Analytics jobs

# Course agenda (*continued #3*)

## Module 7

Orchestrating data movement with  
Azure Data Factory

**Lesson 01** – Explain how Azure Data Factory works

**Lesson 02** – Create linked services and datasets

**Lesson 03** – Create pipelines and activities

**Lesson 04** – Azure Data Factory pipeline execution  
and triggers

## Module 8

Securing Azure Data Platforms

**Lesson 01** – Introduction to security

**Lesson 02** – Key security components

**Lesson 03** – Securing storage accounts and Data Lake  
Storage

**Lesson 04** – Security data stores

**Lesson 05** – Securing streaming data

# Course agenda (*continued #4*)

## Module 9

Monitoring and troubleshooting Data Storage and processing

**Lesson 01** – Explain the monitoring capabilities that are available

---

**Lesson 02** – Troubleshoot common data storage issues

---

**Lesson 03** – Troubleshoot common data processing issues

---

**Lesson 04** – Manage disaster recovery

# Audience

## Primary audience:

The audience for this course are data professionals, data architects, and business intelligence professionals who want to learn about the data platform technologies that exist on Microsoft Azure

## Secondary audience:

The secondary audience for this course are individuals who develop applications that deliver content from the data platform technologies that exist on Microsoft Azure



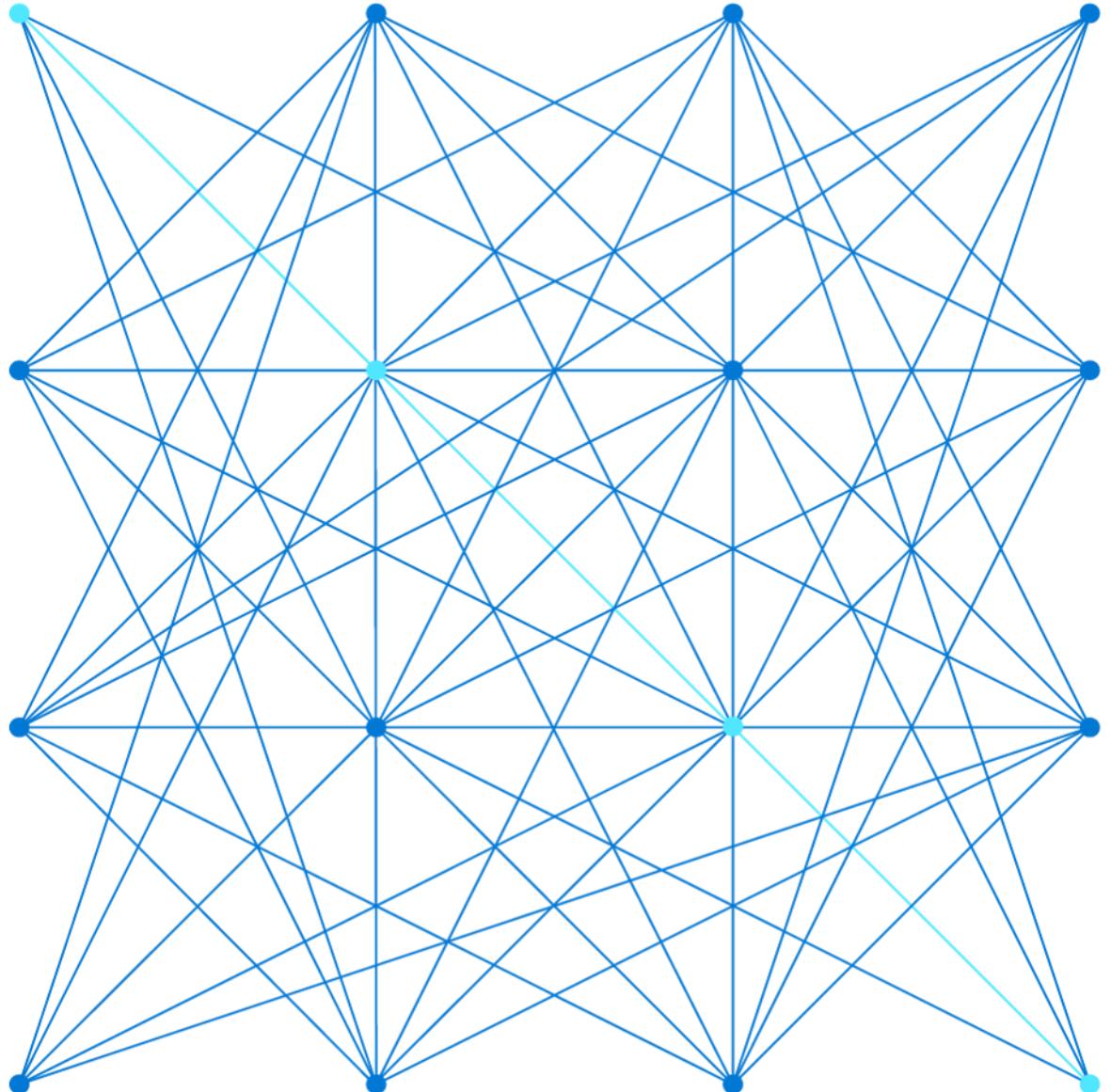
## Prerequisites

In addition to their professional experience, students who take this training should have technical knowledge equivalent to the following courses:

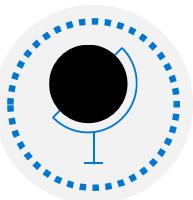
[Azure fundamentals](#)



# Module 01: Azure for the Data Engineer

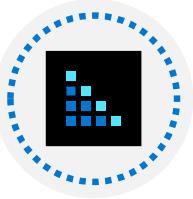


# Agenda



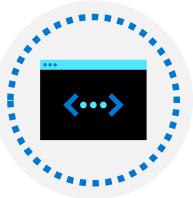
Lesson 01 – Explain the evolving world of data

---



Lesson 02 – Survey the services in the Azure Data Platform

---



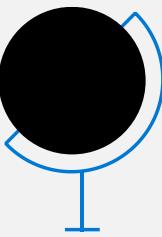
Lesson 03 – Identify the tasks that are performed by a Data Engineer

---

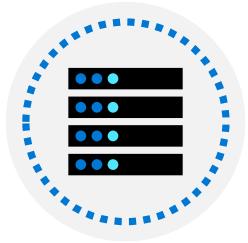


Lesson 04 – Describe the use cases for the cloud in a case study

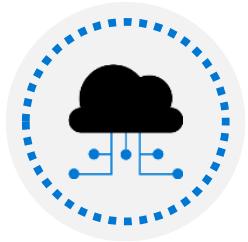
# Lesson 01: The evolving world of data



## Lesson objectives



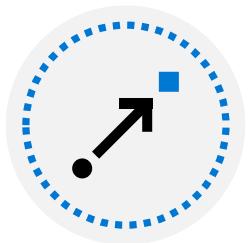
Data abundance



Differences between on-premises and cloud data technologies



How the role of the data professional is changing in organizations



Identify use cases impacted by these changes

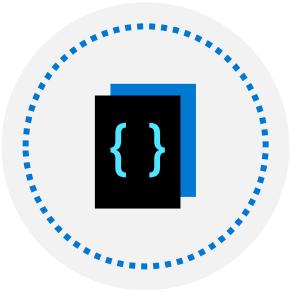
# Data abundance

Processes	Businesses are tasked to store, interpret, manage, transform, process, aggregate and report on data
Consumers	There are a wider range of consumers using different types of devices to consume or generate data
Variety	There's a wider variety of data types that need to be processed and stored
Responsibilities	A data engineer's role is responsible for more data types and technologies
Technologies	Microsoft Azure provides a wide set of tools and technologies

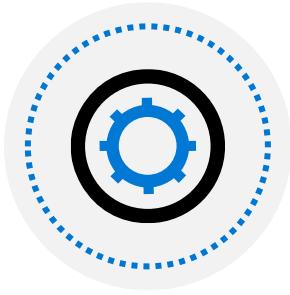
# On-premises versus cloud technologies



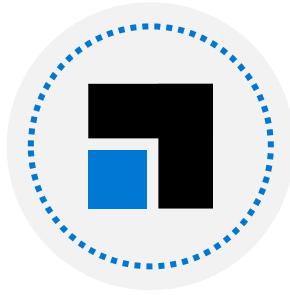
Computing  
Environment



Licensing  
Model



Maintainability

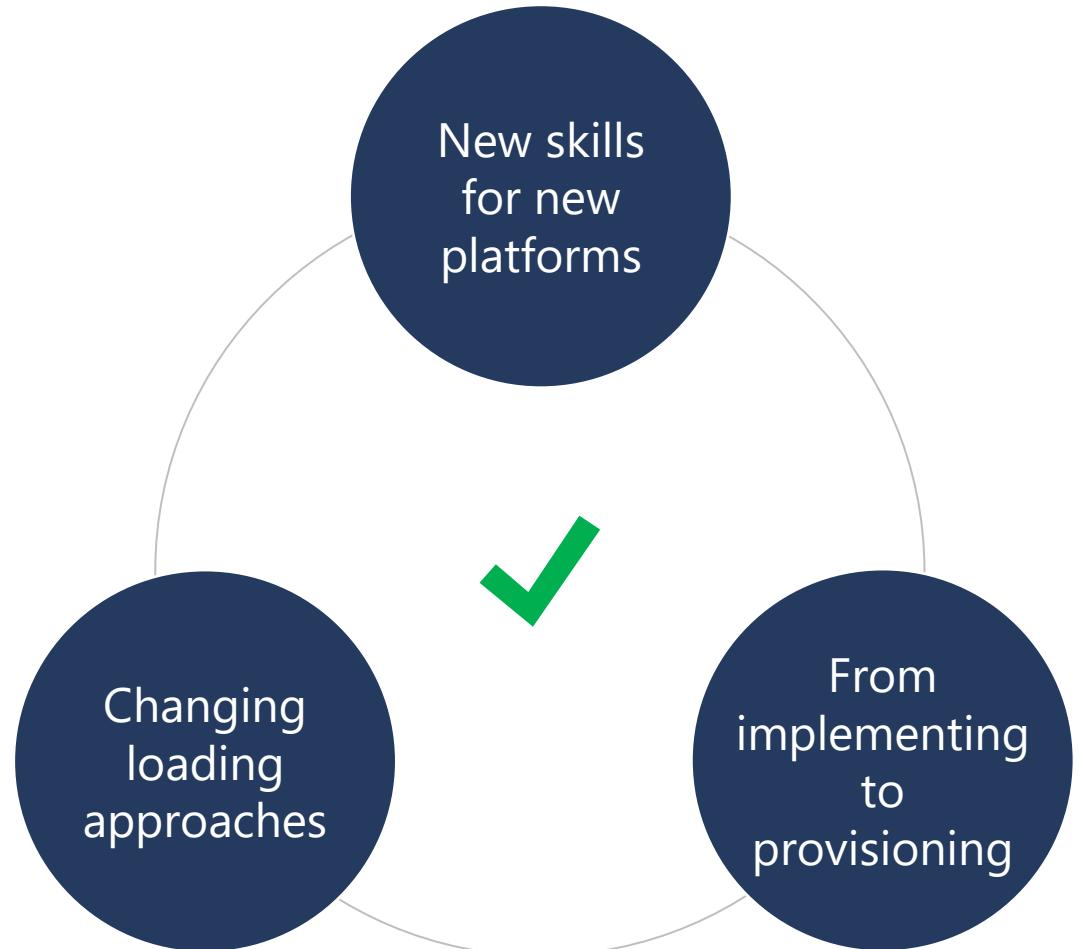


Scalability



Availability

# Data engineering job responsibilities



# Use cases for the cloud

Here are some examples of industries making use of the cloud

## Web retail

Using Azure Cosmos DB's multi-master replication model along with Microsoft's performance commitments, Data Engineers can implement a data architecture to support web and mobile applications that achieve less than a 10-ms response time anywhere in the world

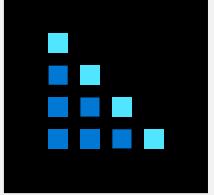
## Healthcare

Azure Databricks can be used to accelerate big data analytics and artificial intelligence (AI) solutions. Within the healthcare industry, it can be used to perform genome studies or pharmacy sales forecasting at petabyte scale

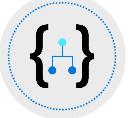
## IoT scenarios

Hundreds of thousands of devices have been designed and sold to generate sensor data known as Internet of Things (IoT) devices. Using technologies like Azure IoT Hub, Data Engineers can easily design a data solution architecture that captures real-time data

## Lesson 02: Survey the services in the Azure Data Platform



# Lesson objectives

-  The differences between structured and unstructured data
-  Azure Storage
-  Azure Data Lake Storage
-  Azure Databricks
-  Azure Cosmos DB
-  Azure SQL Database
-  Azure SQL Data Warehouse
-  Azure Stream Analytics
-  Additional Azure Data Platform Services

# Structured versus Unstructured data

There are three broad types of data and Microsoft Azure provides many data platform technologies to meet the needs of the wide varieties of data

## Structured

Structured data is data that adheres to a schema, so all of the data has the same fields or properties. Structured data can be stored in a database table with rows and columns

## Semi-Structured

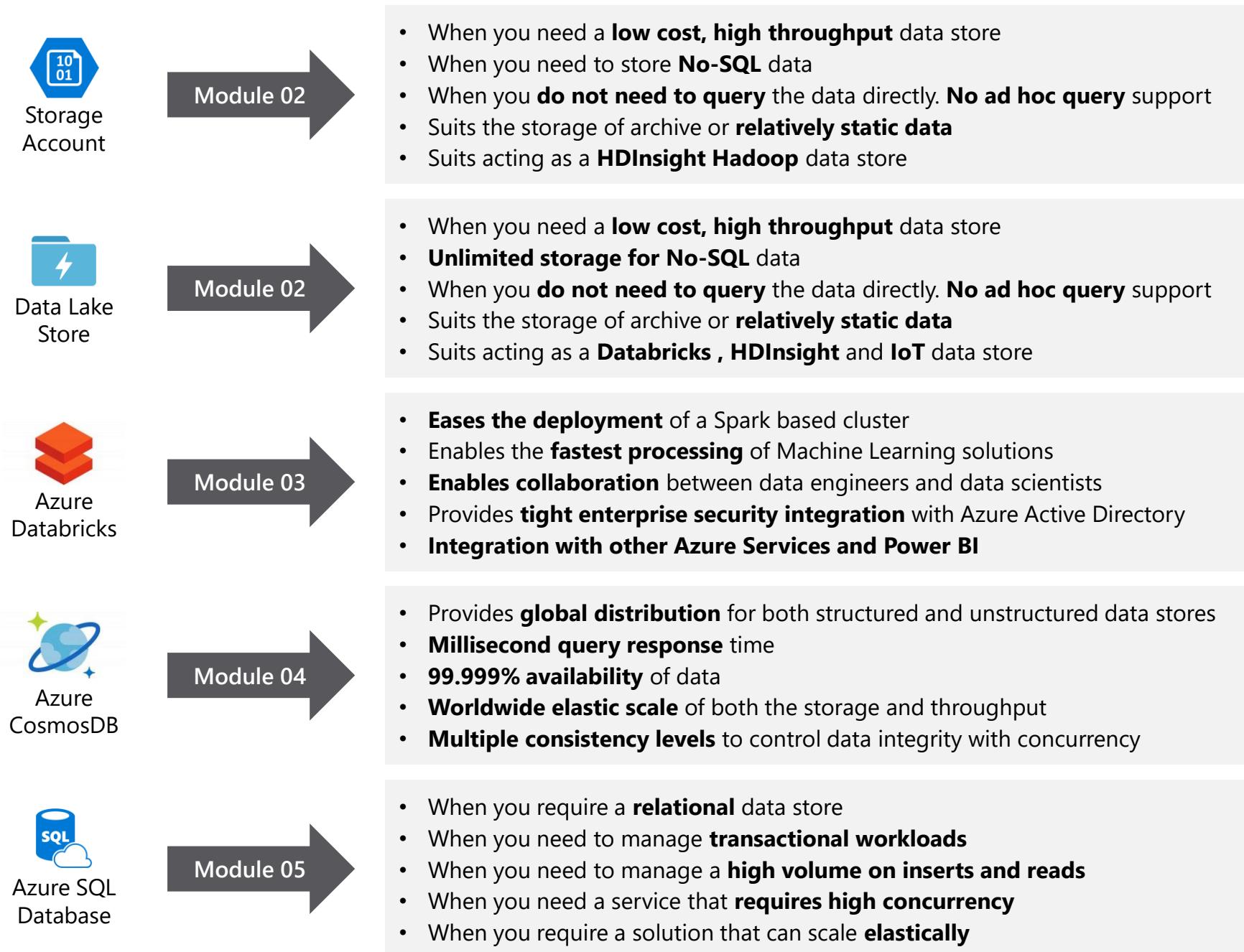
Semi-structured data doesn't fit neatly into tables, rows, and columns. Instead, semi-structured data uses `_tags_` or `_keys_` that organize and provide a hierarchy for the data

## Unstructured

Unstructured data encompasses data that has no designated structure to it. Known as No-SQL, there are four types of No-SQL databases:

- Key Value Store
- Document Database
- Graph Databases
- Column Base

# What to use for Data



# What to use for Data



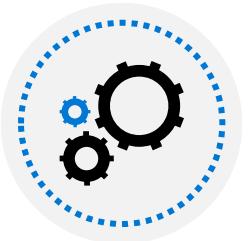
## Lesson 03: Identify the tasks performed by a Data Engineer



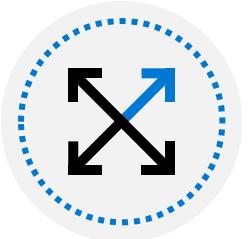
## Lesson objectives



List the new roles of modern data projects

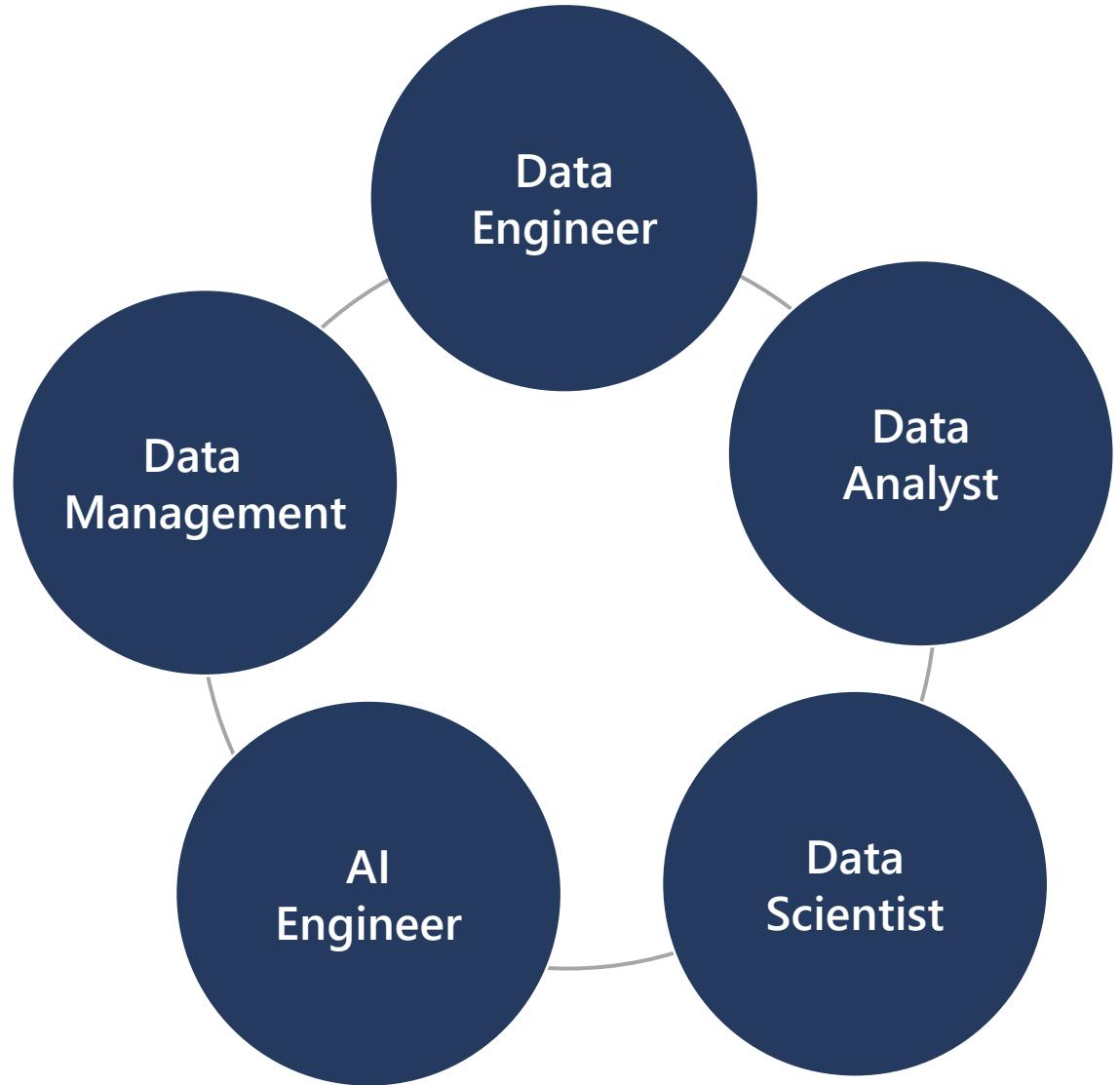


Outline data engineering practices

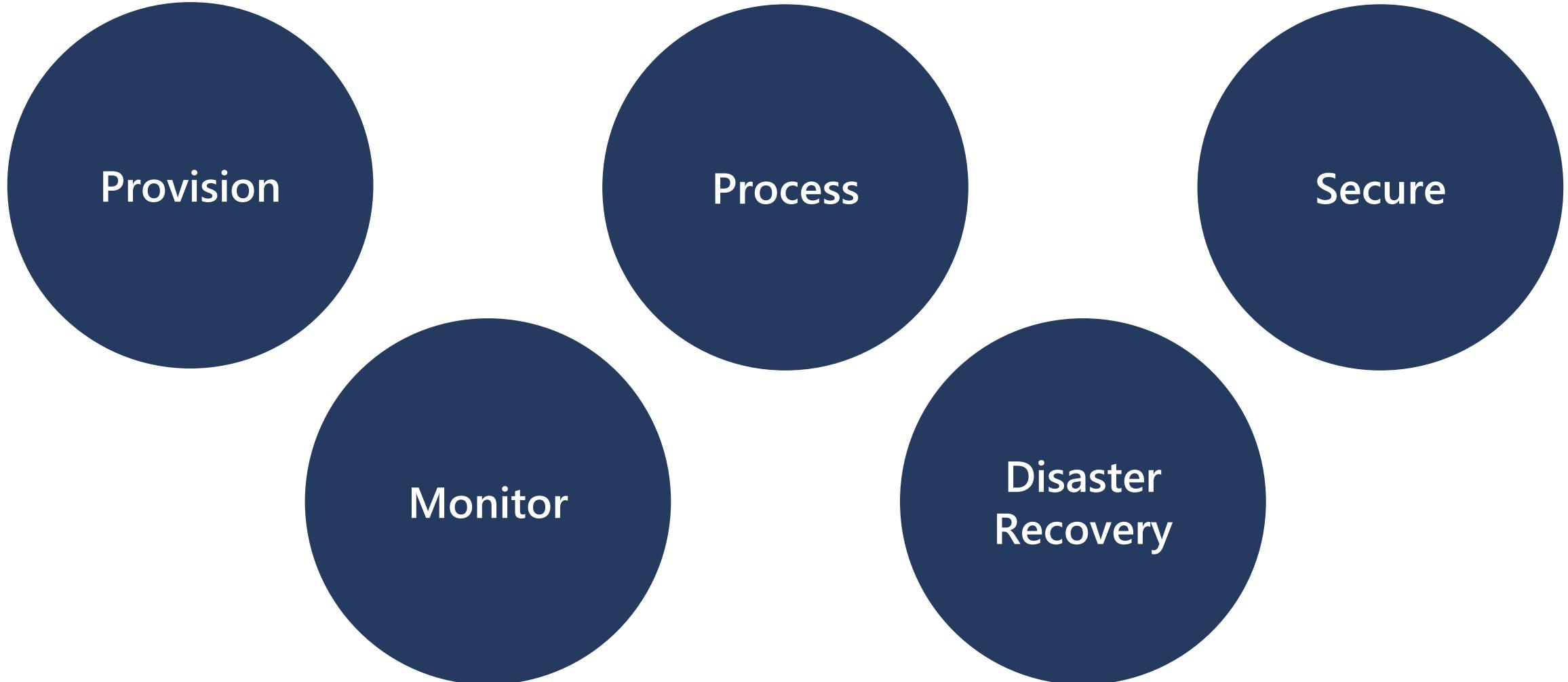


Explore the high-level process for architecting a data engineering project

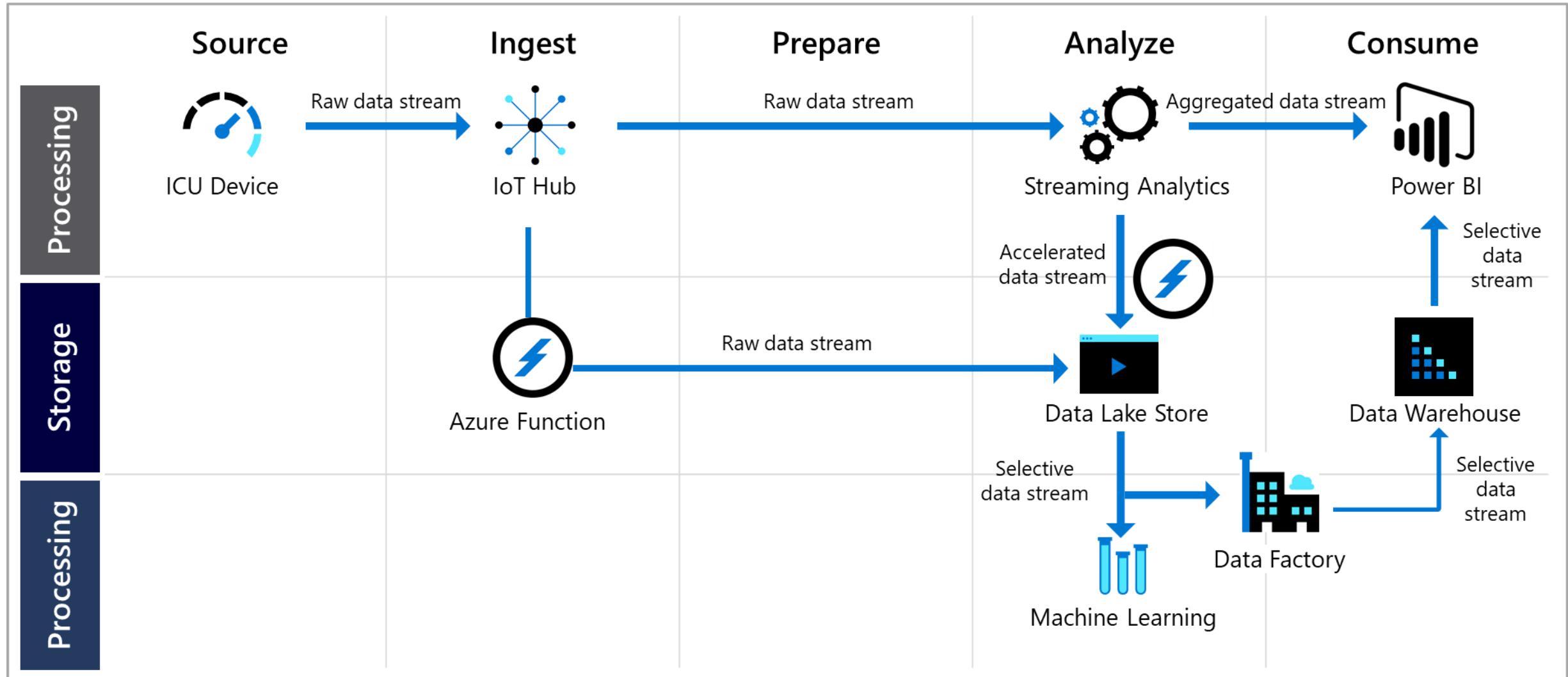
# Roles in data projects



# Data engineering practices



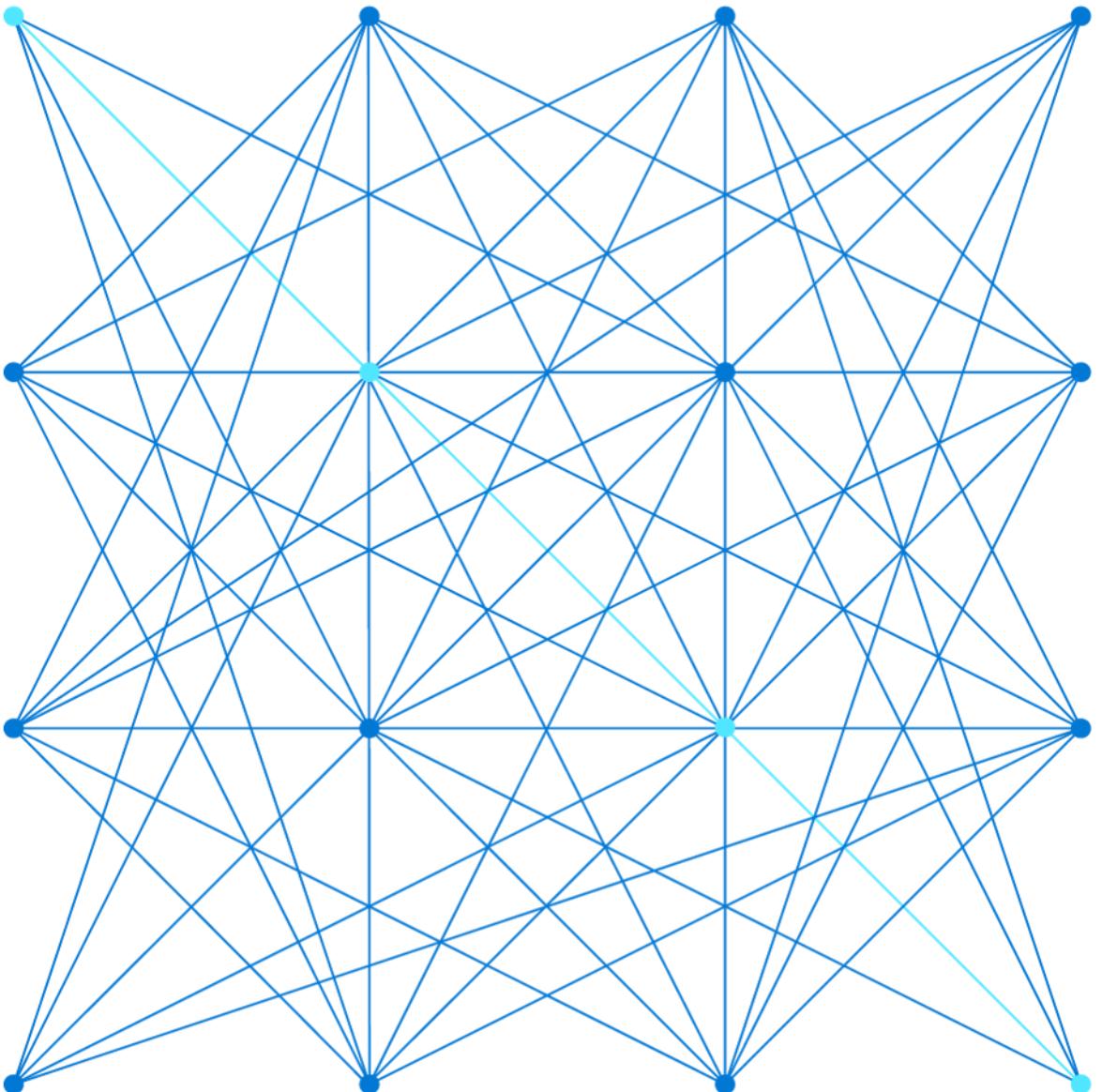
# Architecting projects – An example



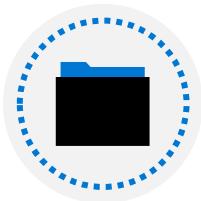


# Module 02:

## Working with data storage



# Agenda



Lesson 01: Choose a data storage approach in Azure

---



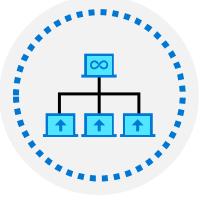
Lesson 02: Create an Azure Storage Account

---



Lesson 03: Explain Azure Data Lake Storage

---

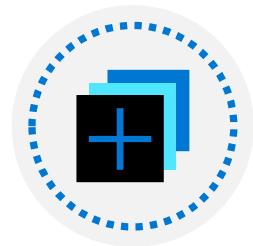


Lesson 04: Upload data into Azure Data Lake Store

# Lesson 01: Choose a data storage approach in Azure



## Lesson objectives



The Benefits of using Azure to store data



Compare Azure data storage with on-premises storage

# Benefits of using Azure to store data



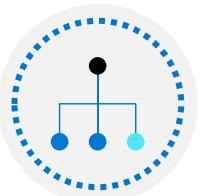
Automated backup



Global replication



Encryption capabilities



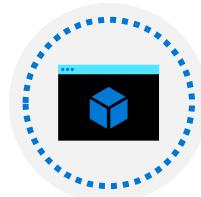
Multiple data types



Support for data analytics



Storage tiers



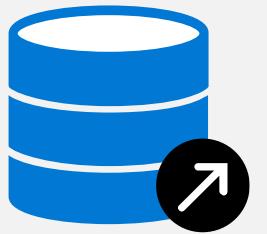
Virtual disks

# Comparing Azure to on-premises storage

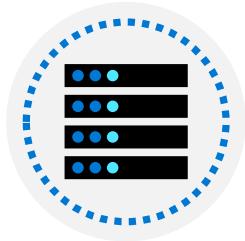
The term “on-premises” refers to the storage and maintenance of data on local hardware and servers

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

## Lesson 02: Create Azure storage account



## Lesson objectives



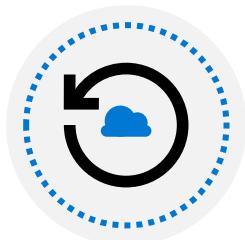
Describe storage accounts



Determine the appropriate settings for each storage account



Choose an account creation tool



Create a storage account using the Azure portal

# 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

Home > New > Storage account > Create storage account

## Create storage account

**Basics** Networking Advanced Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.  
[Learn more about Azure storage accounts](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* chtestao

Resource group \* Select existing...  
[Create new](#)

**Instance details**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name \*

Location \* (US) South Central US

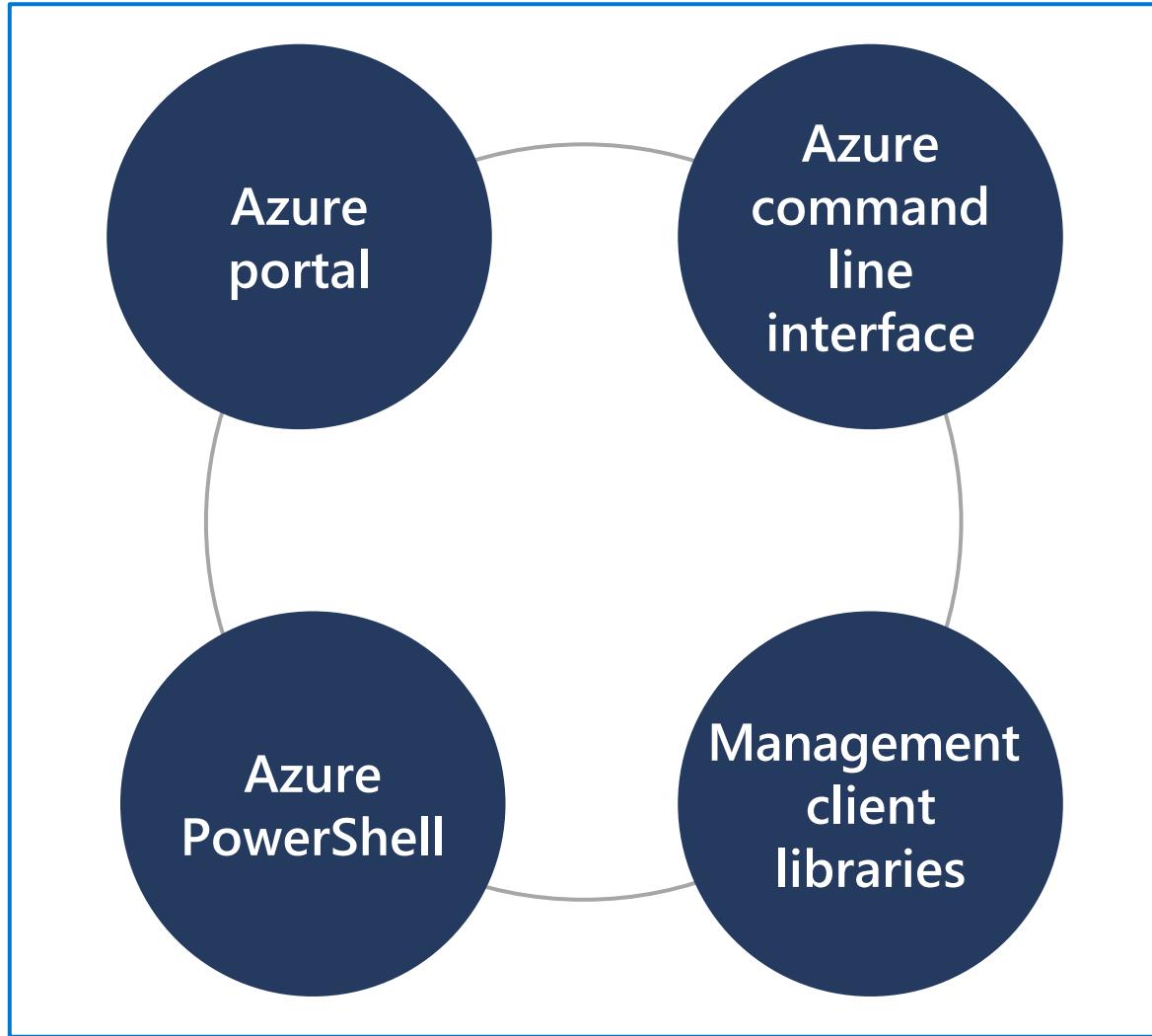
Performance  Standard  Premium

Account kind  StorageV2 (general purpose v2)

Replication  Read-access geo-redundant storage (RA-GRS)

Access tier (default)  Cool  Hot

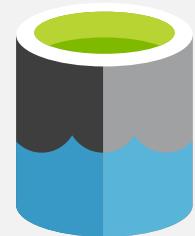
# Storage account creation tool



# Create a storage account



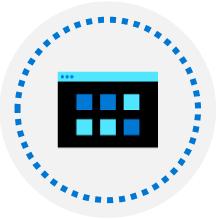
# Lesson 03: Azure Data Lake Storage



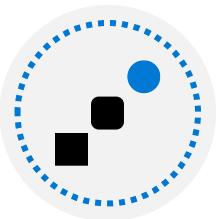
## Lesson objectives



Explain Azure Data Lake Storage



Create an Azure Data Lake Store Gen 2 using the portal



Compare Azure Blob Storage and Data Lake Store Gen 2

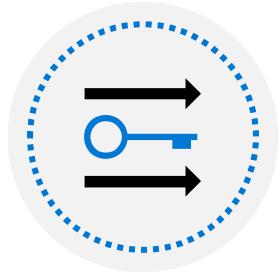


Explore the stages for processing Big Data using Azure Data Lake Store



Describe the use cases for Data lake Storage

# Azure Data Lake Storage – Generation II



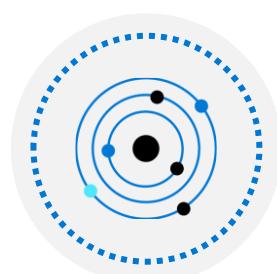
Hadoop access



Security



Performance



Redundancy

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

Home > New > Storage account > Create storage account

## Create storage account

Basics Networking Advanced Tags Review + create

**Security**

Secure transfer required ⓘ  Disabled  Enabled

**Azure Files**

Large file shares ⓘ  Disabled  Enabled

ⓘ The current combination of storage account kind, performance, replication and location does not support large file shares.

**Data protection**

Blob soft delete ⓘ  Disabled  Enabled

ⓘ Data protection and hierarchical namespace cannot be enabled simultaneously.

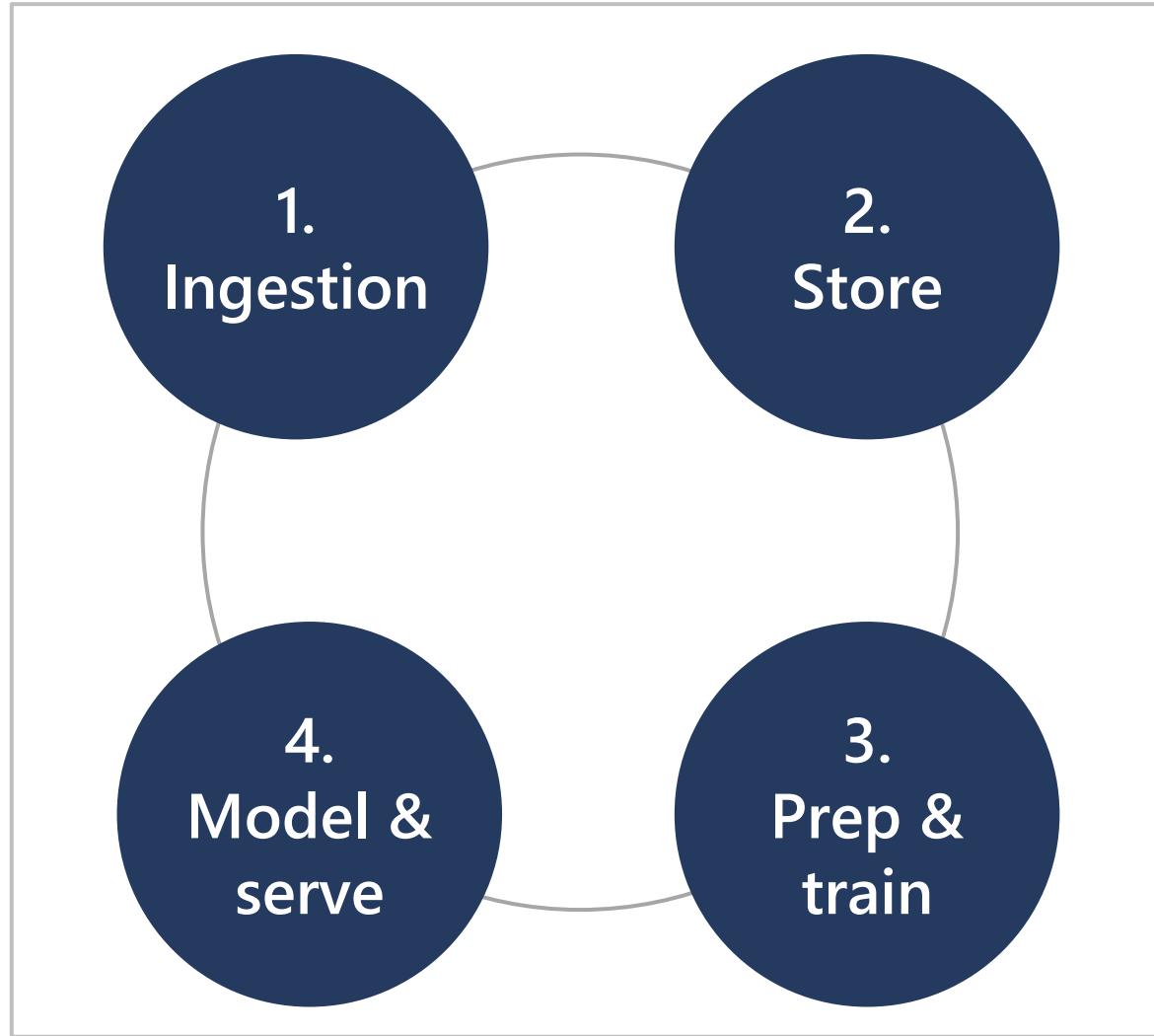
**Data Lake Storage Gen2**

Hierarchical namespace ⓘ  Disabled  Enabled

NFS v3 ⓘ  Disabled  Enabled

Signup is currently required to utilize the NFS v3 feature on a per-subscription basis. [Signup for NFS v3](#)

# Processing Big Data with Azure Data Lake Store



# Big Data use cases

Let's examine three use cases for leveraging an Azure Data Lake Store

## Modern data warehouse

This architecture sees Azure Data Lake Storage at the heart of the solution for a modern data warehouse. Using Azure Data Factory to ingest data into the Data Lake from a business application, and predictive models built in Azure Databricks, using Azure Synapse Analytics as a serving layer

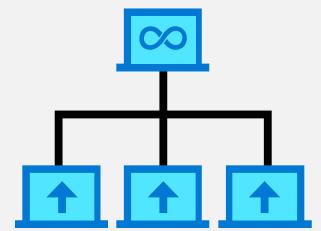
## Advanced analytics

In this solution, Azure Data factory is transferring terabytes of web logs from a web server to the Data Lake on an hourly basis. This data is provided as features to the predictive model in Azure Databricks, which is then trained and scored. The result of the model is then distributed globally using Azure Cosmos DB, that an application uses

## Real time analytics

In this architecture, there are two ingestion streams. Azure Data Factory is used to ingest the summary files that are generated when the HGV engine is turned off. Apache Kafka provides the real-time ingestion engine for the telemetry data. Both data streams are stored in Data Lake store for use in the future

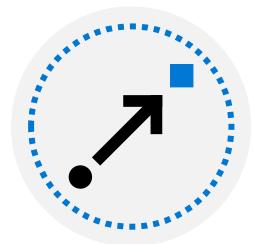
## Lesson 04: Upload data into Azure Data Lake Store



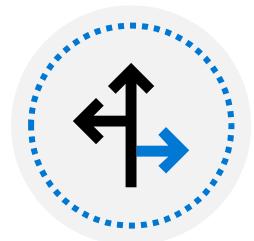
## Lesson objectives



Create an Azure Data Lake Gen2 Store using PowerShell



Upload data into the Data Lake Storage Gen2 using Azure Storage Explorer



Copy data from an Azure Data Lake Store Gen1 to an Azure Data Lake Store Gen2

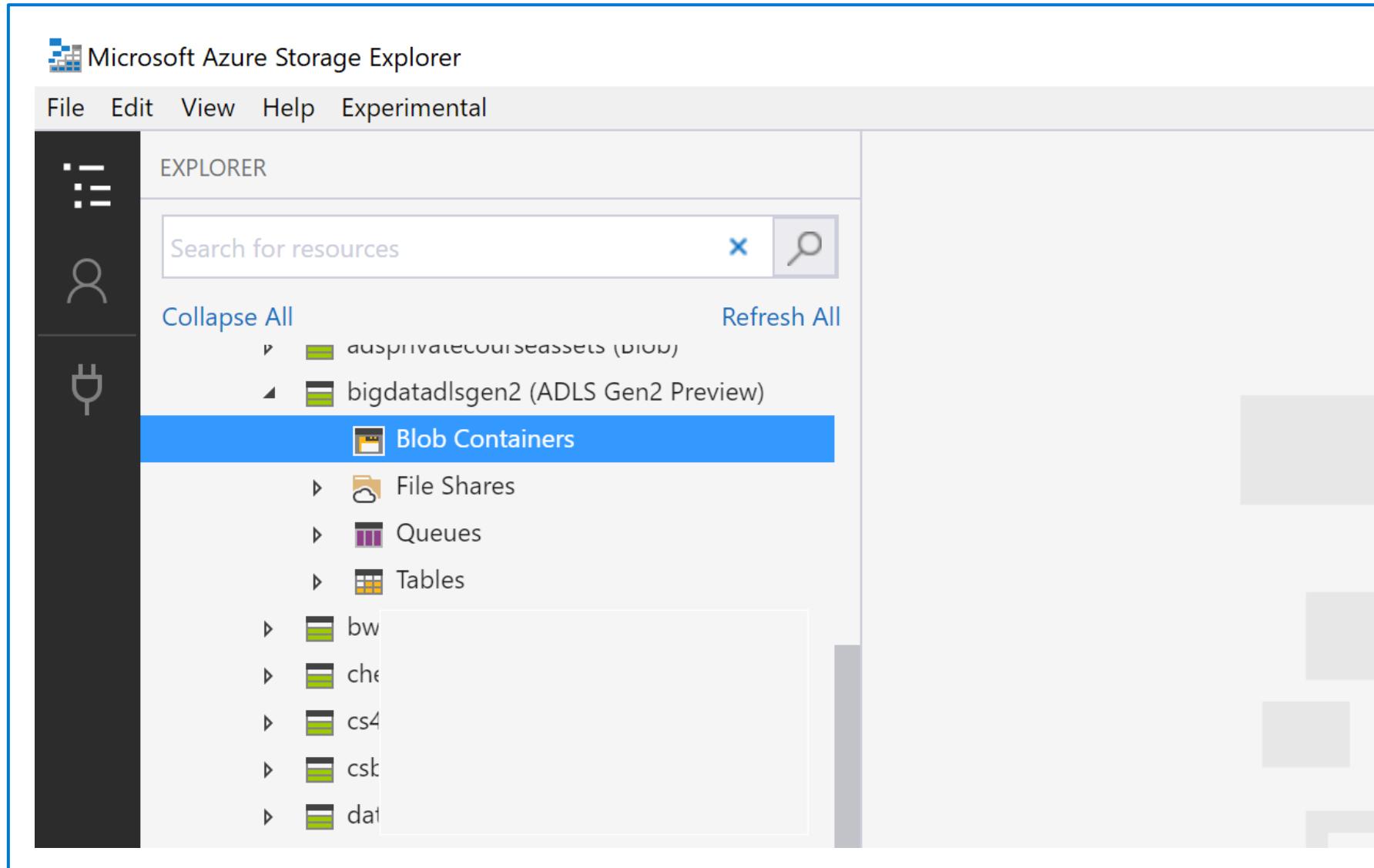
# Create a Azure Data Lake Store (Gen II) using PowerShell

Windows PowerShell

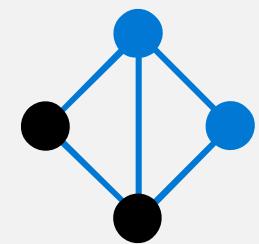
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
PS C:\Users> $location = "westus2"
>>
>> New-AzStorageAccount -ResourceGroupName $resourceGroup
>>   -Name "storagequickstart"
>>   -Location $location
>>   -SkuName Standard_LRS
>>   -Kind StorageV2
>>   -EnableHierarchicalNamespace $True
```

# Uploading data with Azure Storage Explorer

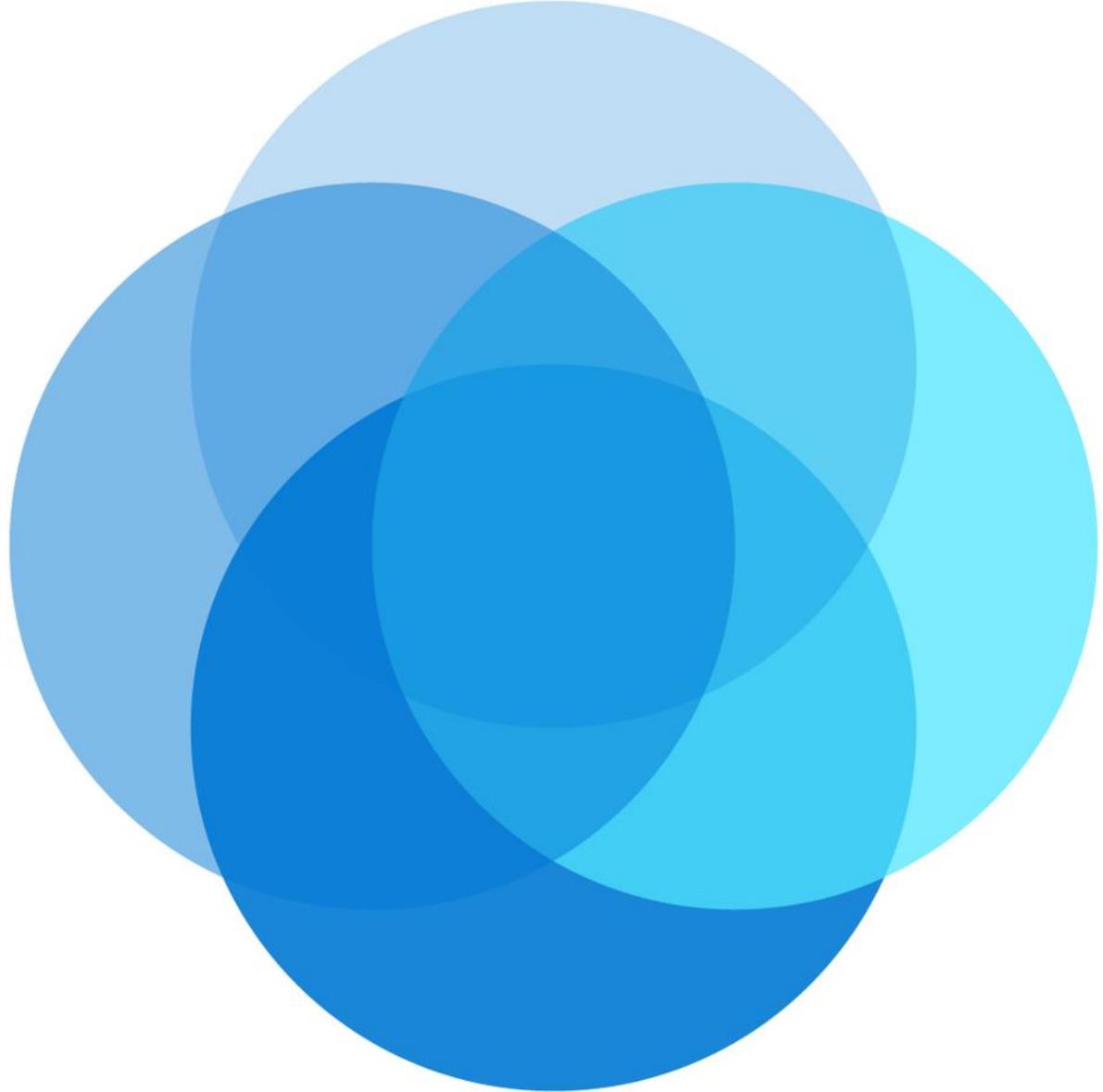


# Lab: Working with data storage





# Module 03: Enabling team based Data Science with Azure Databricks

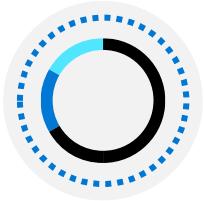


# Agenda



Lesson 01 – Describe Azure Databricks

---



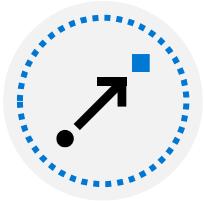
Lesson 02 – Provision Azure Databricks and Workspaces

---



Lesson 03 – Read data using Azure Databricks

---

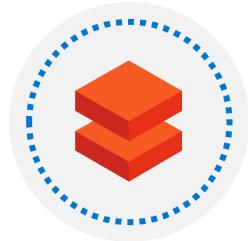


Lesson 04 – Perform transformations with Azure Databricks

# Lesson 01: Describe Azure Databricks



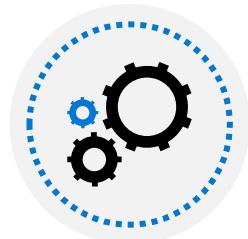
## Lesson objectives



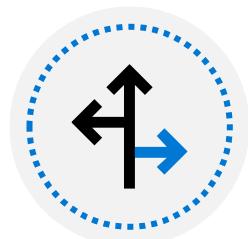
What is Azure Databricks



What are Spark based analytics platform



How Azure Databricks integrates with enterprise security



How Azure Databricks integrates with other cloud services

# What is Azure Databricks



## Apache Spark-based analytics platform:

Simplifies the provisioning and collaboration of Apache Spark-based analytical solutions

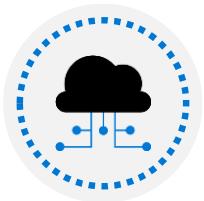
---



## Enterprise Security:

Utilizes the security capabilities of Azure

---



## Integration with other Cloud Services:

Can integrate with a variety of Azure data platform services and Power BI

# What is Apache Spark

Apache Spark emerged to provide a parallel processing framework that supports in-memory processing to boost the performance of big-data analytical applications on massive volumes of data

## Interactive Data Analysis:

Used by business analysts or data engineers to analyze and prepare data

## Streaming Analytics:

Ingest data from technologies such as Kafka and Flume to ingest data in real-time

## Machine Learning:

Contains a number of libraries that enables a Data Scientist to perform Machine Learning

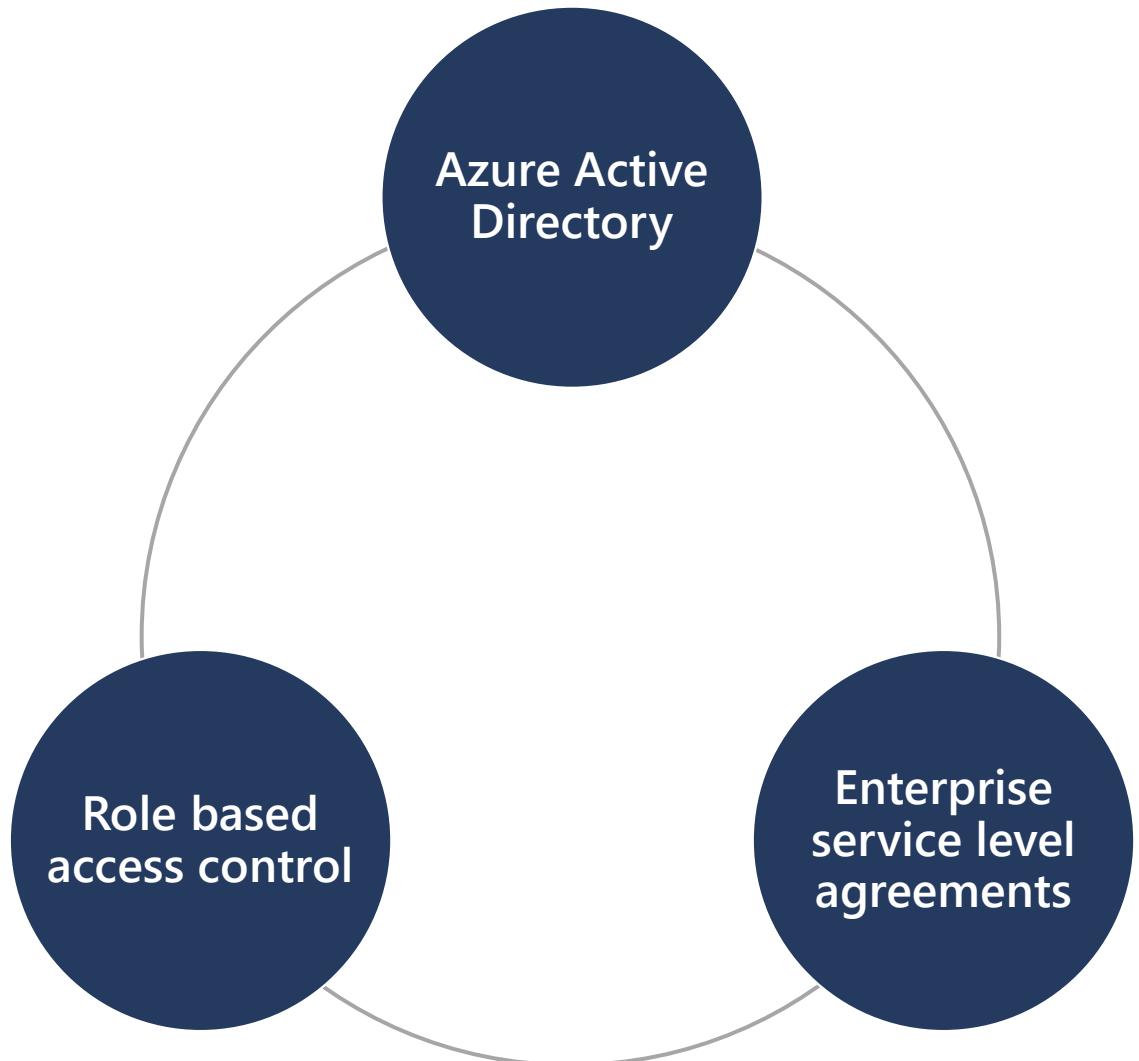
## Why use Azure Databricks?

Azure Databricks is a wrapper around Apache Spark that simplifies the provisioning and configuration of a Spark cluster in a GUI interface

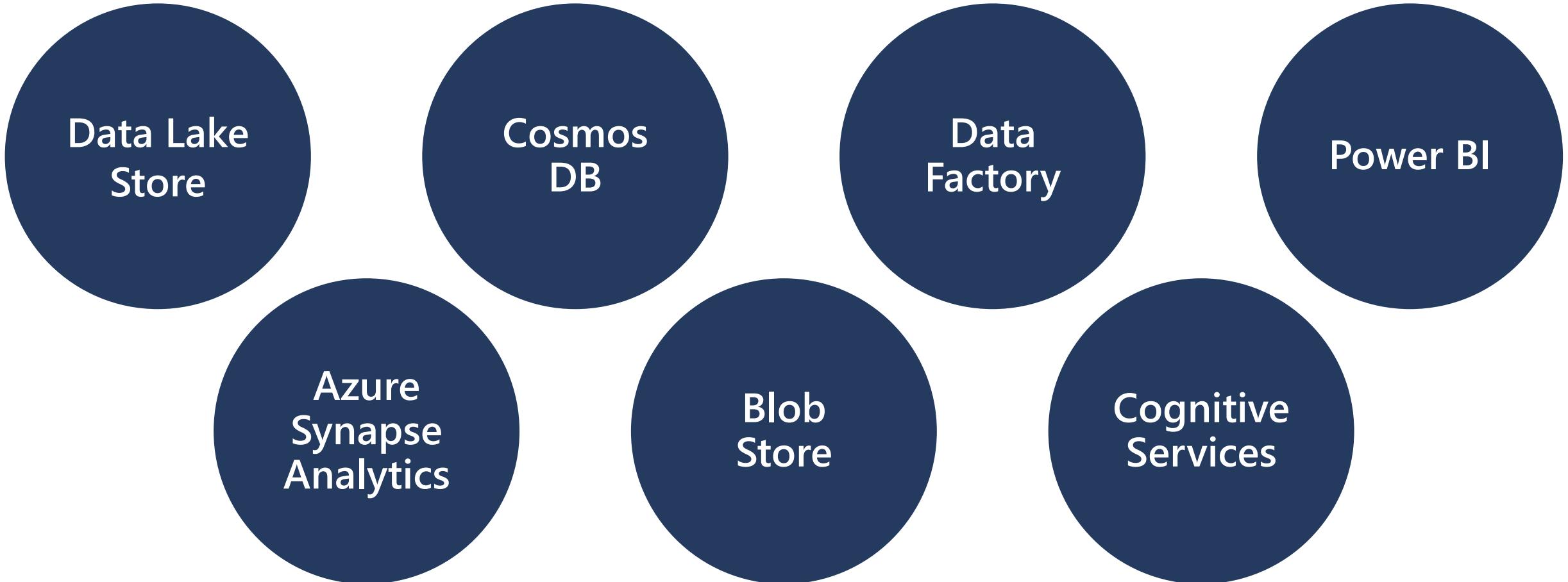
## Azure Databricks components:

Spark SQL and DataFrames  
Streaming  
Mlib  
GraphX  
Spark Core API

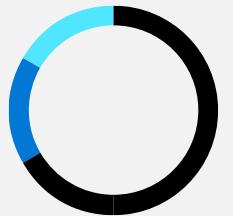
# Enterprise security



# Integration with cloud services



## Lesson 02: Provision Azure Databricks and Workspaces



## Lesson objectives



Create your own Azure Databricks workspace



Create a cluster and notebook in Azure Databricks

# Create an Azure Databricks Workspace

Home > New > Azure Databricks > Azure Databricks Service

## Azure Databricks Service

\* Workspace name: ds-mslearn ✓

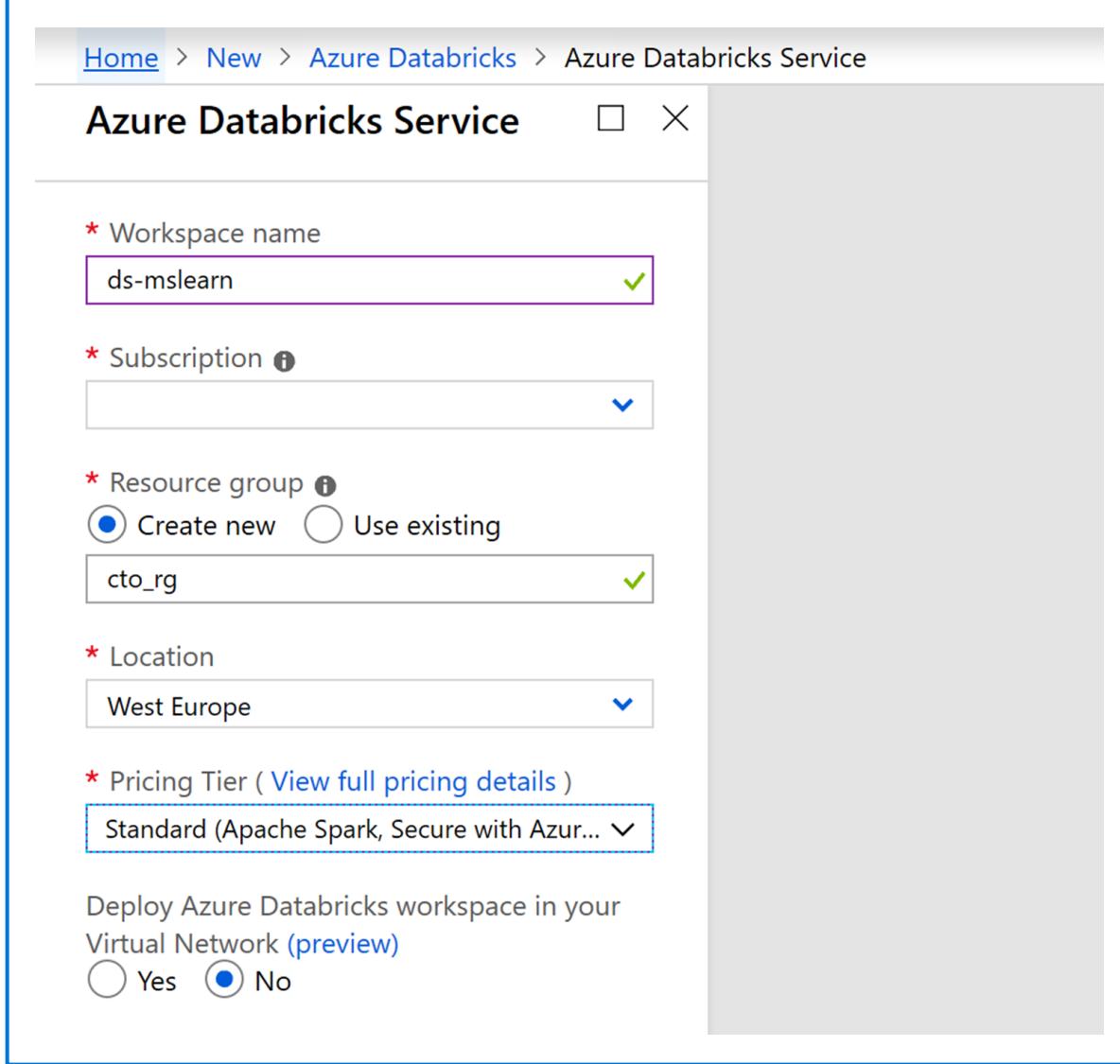
\* Subscription:

\* Resource group:  Create new  Use existing  
cto\_rg ✓

\* Location: West Europe

\* Pricing Tier (View full pricing details): Standard (Apache Spark, Secure with Azur...)

Deploy Azure Databricks workspace in your Virtual Network (preview)  
 Yes  No



# Create a Cluster and Notebook in Azure Databricks

Microsoft Azure PORTAL @microsoft.com

Azure Databricks

Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

Import & Explore Data

Quickly import data, preview its schema, create a table, and query it in a notebook.

Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

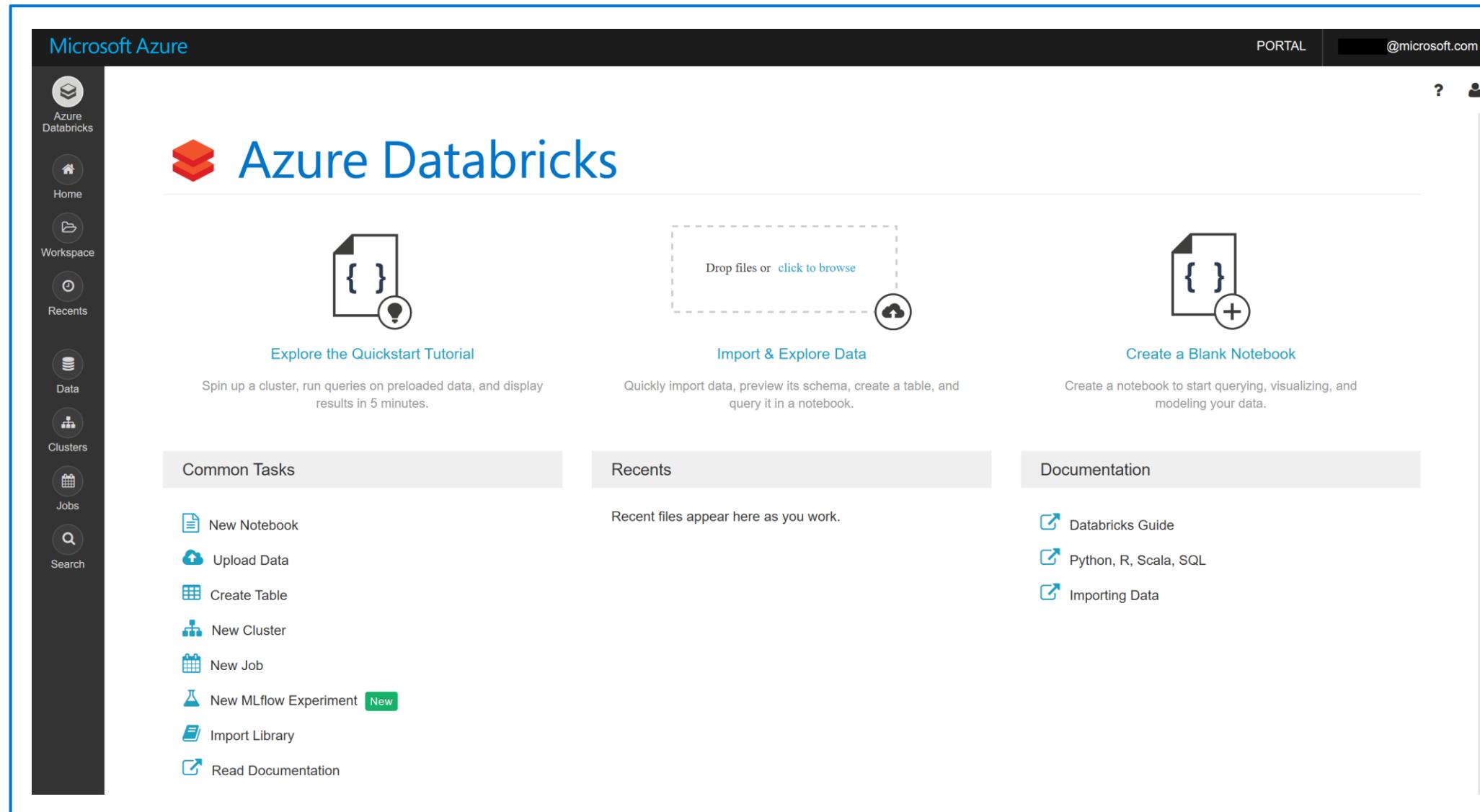
- New Notebook
- Upload Data
- Create Table
- New Cluster
- New Job
- New MLflow Experiment New
- Import Library
- Read Documentation

Recents

Recent files appear here as you work.

Documentation

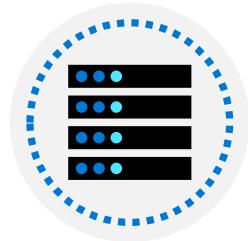
- Databricks Guide
- Python, R, Scala, SQL
- Importing Data



## Lesson 03: Read data using Azure Databricks



## Lesson objectives

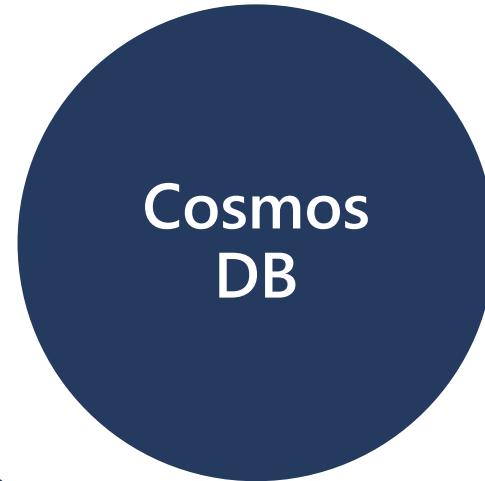


Use Azure Databricks to access data sources



Reading data in Azure Databricks

# Use Azure Databricks to access data sources



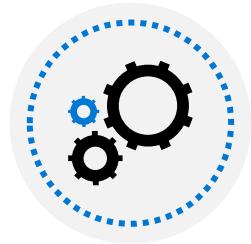
# Reading data in Azure Databricks

SQL	DataFrame (Python)
SELECT col_1 FROM myTable	df.select(col("col_1"))
DESCRIBE myTable	df.printSchema()
SELECT * FROM myTable WHERE col_1 > 0	df.filter(col("col_1") > 0)
..GROUP BY col_2	..groupBy(col("col_2"))
..ORDER BY col_2	..orderBy(col("col_2"))
..WHERE year(col_3) > 1990	..filter(year(col("col_3")) > 1990)
SELECT * FROM myTable LIMIT 10	df.limit(10)
display(myTable) (text format)	df.show()
display(myTable) (html format)	display(df)

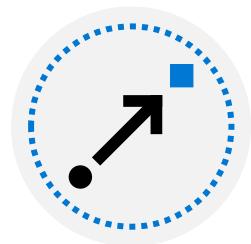
## Lesson 04: Perform transformations with Azure Databricks



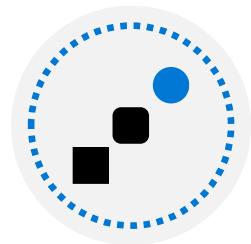
## Lesson objectives



Performing ETL to populate a data model



Perform basic transformations



Perform advanced transformations with user-defined functions

# Performing ETL to populate a data model

The goal of transformation in Extract Transform Load (ETL) is to transform raw data to populate a data model

Extraction	Data validation	Transformation	Corrupt record handling	Loading data
<b>Connect to many data stores:</b> Postgres SQL Server Cassandra Cosmos DB CSV, Parquet Many more..	Validate that the data is what you expect	Applying structure and schema to your data to transform it into the desired format	Built-in functions of Databricks allow you to handle corrupt data such as missing and incomplete information	Highly effective design pattern involves loading structured data back to DBFS as a parquet file

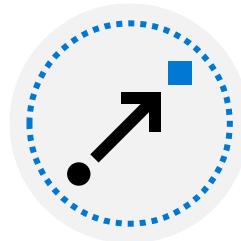
## Basic transformation



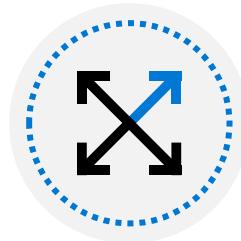
Normalizing values



Missing/null data



De-duplication



Pivoting data frames

# Advanced transformations

Advanced data transformation using custom and advanced user-defined functions, managing complex tables and loading data into multiple databases simultaneously

## User-defined functions

This fulfils scenarios when you need to define logic specific to your use case and when you need to encapsulate that solution for reuse. UDFs provide custom, generalizable code that you can apply to ETL workloads when Spark's built-in functions won't suffice

## Joins and lookup tables

A standard (or shuffle) join moves all the data on the cluster for each table to a given node on the cluster. This is an expensive operation. Broadcast joins remedy this situation when one DataFrame is sufficiently small enough to duplicate on each node of the cluster, avoiding the cost of shuffling a bigger DataFrame

## Multiple databases

Loading transformed data to multiple target databases can be a time-consuming activity. Partitions and slots are options to get optimum performance from database connections. A partition refers to the distribution of data while a slot refers to the distribution of computation

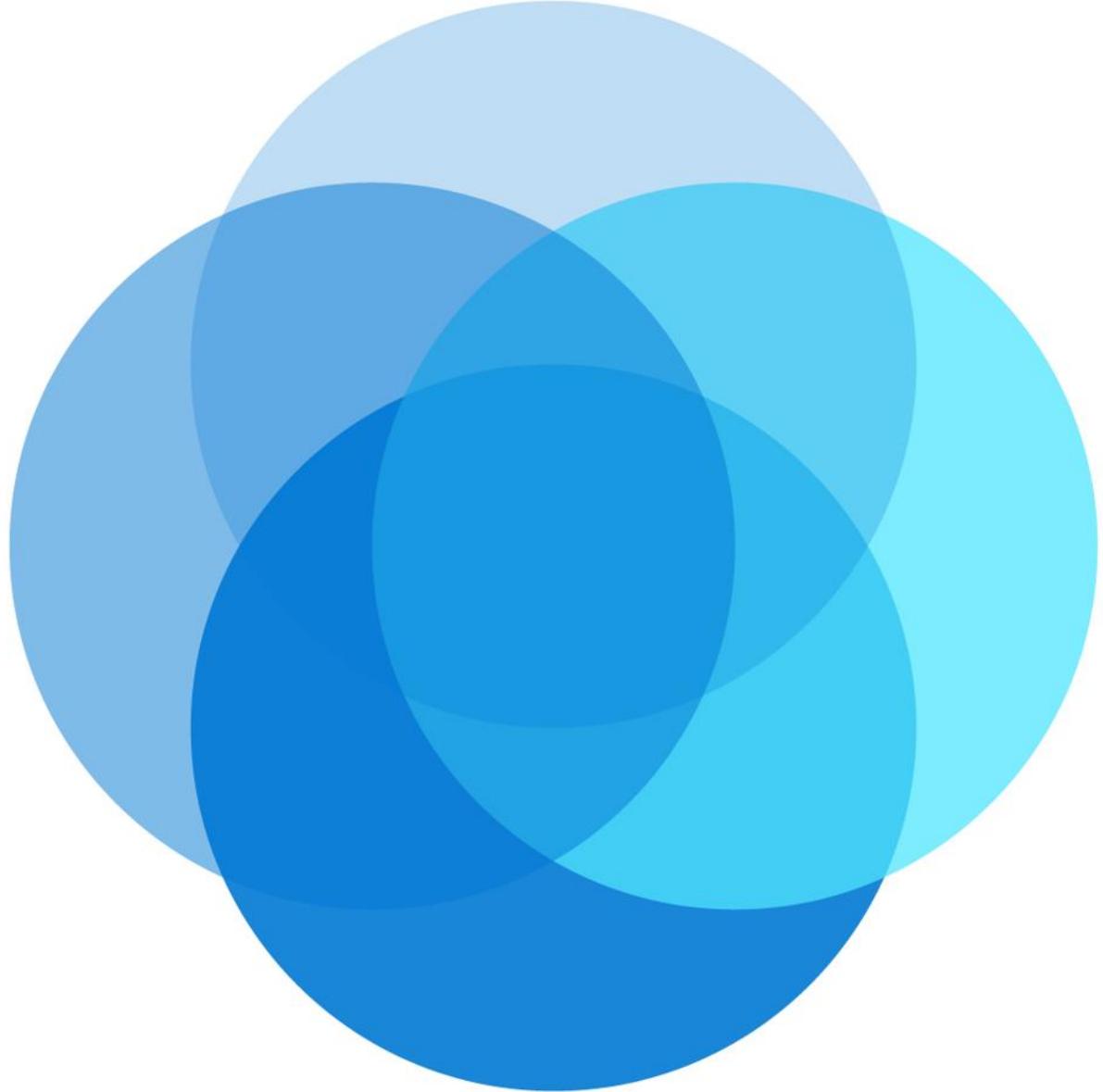
# Lab: Enabling team-based Data Science with Azure Databricks



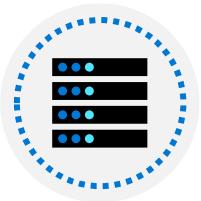


# Module 04:

## Building globally distributed databases with Cosmos DB

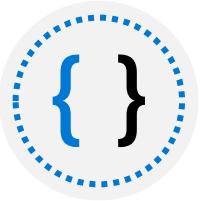


# Agenda



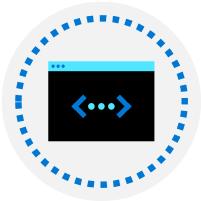
Lesson 01: Create an Azure Cosmos DB database built to scale

---



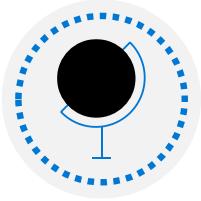
Lesson 02: Insert and query data in your Azure Cosmos DB database

---



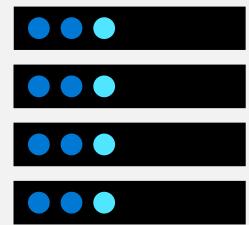
Lesson 03: Build a .NET Core app for Azure Cosmos DB in Visual Studio Code

---



Lesson 04: Distribute your data globally with Azure Cosmos DB

## Lesson 01: Create an Azure Cosmos DB database built to scale



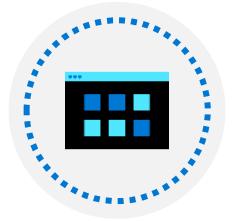
## Lesson objectives



What is Cosmos DB



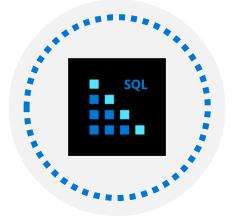
Create an Azure Cosmos DB account



What is a Request Unit

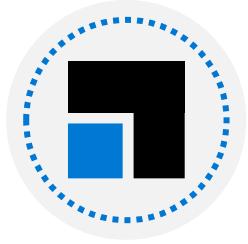


Choose a partition key



Create a database and container for NoSQL data in Azure Cosmos DB

# What is Azure Cosmos DB



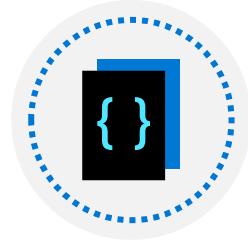
Scalability



Performance



Availability



Programming  
model

# Create an Azure Cosmos DB account

Home > New Create Azure Cosmos DB Account

## Create Azure Cosmos DB Account

**Basics** Networking Tags Review + create

Azure Cosmos DB is a globally distributed, multi-model, fully managed database service. [Try it for free](#), for 30 days with unlimited renewals. Go to production starting at \$24/month per database, multiple containers included. [Learn more](#)

**Project Details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

**Subscription \*** chtestao

**Resource Group \*** Select existing... Create new

**Instance Details**

**Account Name \*** Enter account name

**API \*** Core (SQL)

**Apache Spark** Notebooks Notebooks with Apache Spark None Sign up for Apache Spark preview

**Location \*** (US) West US

**Geo-Redundancy** Enable Disable

**Multi-region Writes** Enable Disable

\*Up to 33% off multi-region writes is available to qualifying new accounts only. Accounts must be created between December 1, 2019 and February 29, 2020. Offer limited to accounts with both account locations and geo-redundancy, and applies only to multi-region writes in those same regions. Both Geo-Redundancy and Multi-region Writes must be enabled in account settings. Actual discount will vary based on number of qualifying regions selected.

# What are Request Units

Throughput is important to ensure you can handle the volume of transactions you need

## Database throughput

Database throughput is the number of reads and writes that your database can perform in a single second

## What is a Request Unit

Azure Cosmos DB measures throughput using something called a request unit (RU). Request unit usage is measured per second, so the unit of measure is request units per second (RU/s). You must reserve the number of RU/s you want Azure Cosmos DB to provision in advance

## Exceeding throughput limits

If you don't reserve enough request units, and you attempt to read or write more data than your provisioned throughput allows, your request will be rate-limited

Item size	Reads/second	Writes/second	Request units
1 KB	500	100	$(500 * 1) + (100 * 5) = 1,000 \text{ RU/s}$
1 KB	500	500	$(500 * 1) + (500 * 5) = 3,000 \text{ RU/s}$
4 KB	500	100	$(500 * 1.3) + (100 * 7) = 1,350 \text{ RU/s}$
4 KB	500	500	$(500 * 1.3) + (500 * 7) = 4,150 \text{ RU/s}$
64 KB	500	100	$(500 * 10) + (100 * 48) = 9,800 \text{ RU/s}$
64 KB	500	500	$(500 * 10) + (500 * 48) = 29,000 \text{ RU/s}$

# Choosing a Partition Key

## Why have a Partition Strategy?

Having a partition strategy ensures that when your database needs to grow, it can do so easily and continue to perform efficient queries and transactions

## What is a Partition Key?

A partition key is the value by which azure organizes your data into logical divisions

# Creating a Database and a Container in Cosmos DB

Add Container X

**Start at \$24/mo per database, multiple containers included** [More details](#)

\* Database id i  
 Create new  Use existing  
Type a new database id

Provision database throughput i

\* Throughput (400 - 100,000 RU/s) i  
 Autopilot (preview)  Manual  
400

Estimated spend (USD): **\$0.032 hourly / \$0.77 daily** (1 region,  
400RU/s, \$0.00008/RU)

\* Container id i  
e.g., Container1

\* Partition key i  
e.g., /address/zipCode

My partition key is larger than 100 bytes

Unique keys i  
+ Add unique key

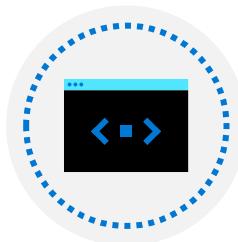
## Lesson 02: Insert and Query Data in your Azure Cosmos DB Database

{ }

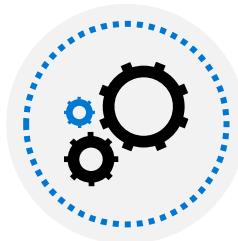
# Lesson objectives



**Create a product catalog document in the Data Explorer:**  
Add data



**Perform Azure Cosmos DB queries:**  
Query types  
Run queries



**Running complex operations on your data**



**Working with graph data**

# Create a product catalog documents in the Data Explorer

The screenshot shows the Azure Cosmos DB Data Explorer interface for the database account "awcdbstudcto". The left sidebar menu is visible, with "Data Explorer" selected. The main area displays a hierarchical tree view under the "Products" category, specifically the "Clothing" subcategory. A table titled "Items" is shown with columns "id" and "...". The first item has an id of 1 and a value of 332... (truncated). To the right of the table, a JSON document is displayed, representing the details of the first product item:

```
{  
  "id": "1",  
  "productId": "33218896",  
  "category": "Women's Clothing",  
  "manufacturer": "Contoso Sport",  
  "description": "Quick dry crew neck t-shirt",  
  "price": "14.99",  
  "shipping": {  
    "weight": 1,  
    "dimensions": {  
      "width": 6,  
      "height": 8,  
      "depth": 1  
    }  
  },  
  "_rid": "P01tAMk-JP8BAAAAAAA=",  
  "_self": "dbs/P01tAA=/colls/P01tAMk-JP8=/docs/1",  
  "_etag": "\\"4100b36e-0000-0d00-0000-5d38cafe0"  
}
```

# Perform Azure Cosmos DB Queries

## SELECT Query Basics

```
SELECT <select_list>
[FROM <optional_from_specification>]
[WHERE <optional_filter_condition>]
[ORDER BY <optional_sort_specification>]
[JOIN <optional_join_specification>]
```

## Examples

```
SELECT *
FROM Products p WHERE p.id ="1"
SELECT p.id, p.manufacturer, p.description
FROM Products p WHERE p.id ="1"
SELECT p.price, p.description, p.productId
FROM Products p ORDER BY p.price ASC
SELECT p.productId
FROM Products p JOIN p.shipping
```

# Running complex operations on data

Multiple documents in your database frequently need to be updated at the same time. The way to perform these transactions in Azure Cosmos DB is by using stored procedures and user-defined functions (UDFs)

## Stored procedures

Stored procedures perform complex transactions on documents and properties. Stored procedures are written in JavaScript and are stored in a collection on Azure Cosmos DB

## User defined functions

User Defined Functions are used to extend the Azure Cosmos DB SQL query language grammar and implement custom business logic, such as calculations on properties and documents

# Working with Graph Data

```
from gremlin_python.driver import client,
serializer
import sys, traceback

CLEANUP_GRAPH = "g.V().drop()"

INSERT_NATIONAL_PARK_VERTICES = [
    "g.addV('Park').property('id',
'p1').property('name',
'Yosemite').property('Feature', 'El Capitan')",
    "g.addV('Park').property('id',
'p2').property('name', 'Joshua
Tree').property('Feature', 'Yucca Brevifolia')",
    "g.addV('State').property('id',
's1').property('name',
'California').property('Location', 'USA')",
    "g.addV('Ecosystem').property('id',
'e1').property('name', 'Alpine')",
    "g.addV('Ecosystem').property('id',
'e2').property('name', 'Desert')",
    "g.addV('Ecosystem').property('id',
'e3').property('name', 'High Altitude')"
]

INSERT_NATIONAL_PARK_EDGES = [
    "g.V('p1').addE('is in').to(g.V('s1'))",
    "g.V('p2').addE('is in').to(g.V('s1'))",
    "g.V('p1').addE('has ecosystem
of').to(g.V('e1'))",
    "g.V('p2').addE('has ecosystem
of').to(g.V('e2'))",
    "g.V('p1').addE('has ecosystem
of').to(g.V('e3'))",
    "g.V('p2').addE('has ecosystem
of').to(g.V('e3'))"
]
```

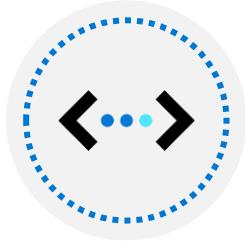
## Lesson 03: Build a .NET Core App for Azure Cosmos DB in VS Code



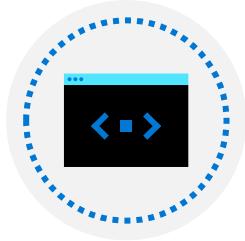
## Lesson objectives



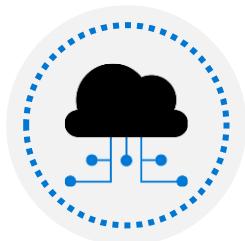
Create an Azure Cosmos DB account, database, and container in Visual Studio Code using the Azure Cosmos DB extension



Create an application to store and query data in Azure Cosmos DB



Use the Terminal in Visual Studio Code to quickly create a console application



Add Azure Cosmos DB functionality with the help of the Azure Cosmos DB extension for Visual Studio Code

# Creating Azure Cosmos DB in Visual Studio Code

The screenshot shows the Visual Studio Code interface with the Azure extension installed. The left sidebar has icons for File Explorer, Search, Task List, and others. The main area shows the Azure service navigation pane with 'AZURE' selected, displaying a database named 'Retail'. Below it, under 'COSMOS DB', are 'adventbikes (MongoDB)', 'ctocdb (SQL)' (selected), 'Products' (under SQL), 'Clothing' (under Products), 'Documents' (under Clothing), and 'Attached Database Accounts'. A search bar above the tree view says 'Press 'Enter' to confirm your input or 'Escape' to cancel'. To the right is a code editor window titled '1-cosmos-document.json - Visual Studio Code' containing a JSON document:

```
1   {
2     "productId": "33218896",
3     "category": "Women's Clothing",
4     "manufacturer": "Contoso Sport",
5     "description": "Quick dry crew neck t-shirt",
6     "price": "14.99",
7     "shipping": {
8       "weight": 1,
9       "dimensions": {
10         "width": 6,
11         "height": 8,
12         "depth": 1
13       }
14     },
15     "_rid": "QSl9ALDRxXUBAAAAAA==",
16     "_self": "dbs/QSl9AA==/colls/QSl9ALDRxXU=/docs",
17     "_etag": "\"13000b6a-0000-0700-0000-5c9b6190",
18     "_attachments": "attachments/",
19     "_ts": 1553686941
20   }
```

# Working with documents programmatically

CreateDocument  
Async

ReadDocument  
Async

ReplaceDocument  
Async

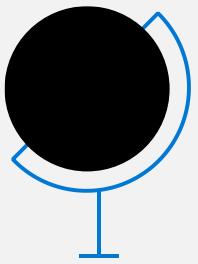
UpsertDocument  
Async

DeleteDocument  
Async

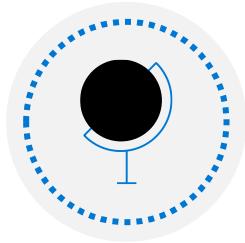
# Querying document programmatically

```
{  
    // Set some common query options  
    FeedOptions queryOptions = new FeedOptions { MaxItemCount = -1, EnableCrossPartitionQuery = true };  
  
    // Here we find nelapin via their LastName  
    IQueryable<User> userQuery = this.client.CreateDocumentQuery<User>(  
        UriFactory.CreateDocumentCollectionUri(databaseName, collectionName), queryOptions)  
        .Where(u => u.LastName == "Pindakova");  
  
    // The query is executed synchronously here, but can also be executed asynchronously via the IDocumentQuery<T> interface  
    Console.WriteLine("Running LINQ query...");  
    foreach (User in userQuery)  
    {  
        Console.WriteLine("\tRead {0}", user);  
    }  
  
    // Now execute the same query via direct SQL  
    IQueryable<User> userQueryInSql = this.client.CreateDocumentQuery<User>(  
        UriFactory.CreateDocumentCollectionUri(databaseName, collectionName),  
        "SELECT * FROM User WHERE User.lastName = 'Pindakova'", queryOptions );  
  
    Console.WriteLine("Running direct SQL query...");  
    foreach (User in userQueryInSql)  
    {  
        Console.WriteLine("\tRead {0}", user);  
    }  
  
    Console.WriteLine("Press any key to continue ...");  
    Console.ReadKey();  
}
```

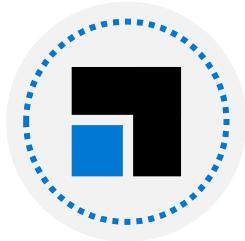
## Lesson 04: Distribute your data globally with Azure Cosmos DB



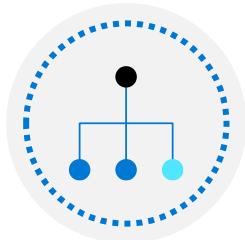
## Lesson objectives



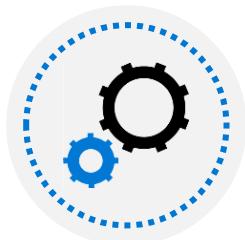
Learn about the benefits of writing and replicating data to multiple regions around the world



Cosmos DB multi-master replication



Cosmos DB failover management



Change the consistency setting for your database

# Benefits of writing and replicating data to multiple regions

Home > Resource groups > cto\_rg > ctoedb > Replicate data globally

## Replicate data globally

ctoedb

Save Discard Manual Failover Automatic Failover

Click on a location to add or remove regions from your Azure Cosmos DB account.

\* Each region is billable based on the throughput and storage for the account. [Learn more](#)



Configure regions

Configure the regions available for reads and writes. [+ Add region](#)

REGIONS	READS ENABLED	WRITES ENABLED	
West US	✓	✓	trash
UK South	✓	✓	trash
Japan West	✓	✓	trash
South Africa North	✓	✓	trash

# Cosmos DB multi-master replication



# Cosmos DB failover management

Automated fail-over is a feature that comes into play when there's a disaster or other event that takes one of your read or write regions offline, and it redirects requests from the offline region to the next most prioritized region

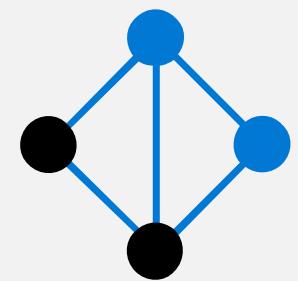
## Read region outage

Azure Cosmos DB accounts with a read region in one of the affected regions are automatically disconnected from their write region and marked offline

## Write region outage

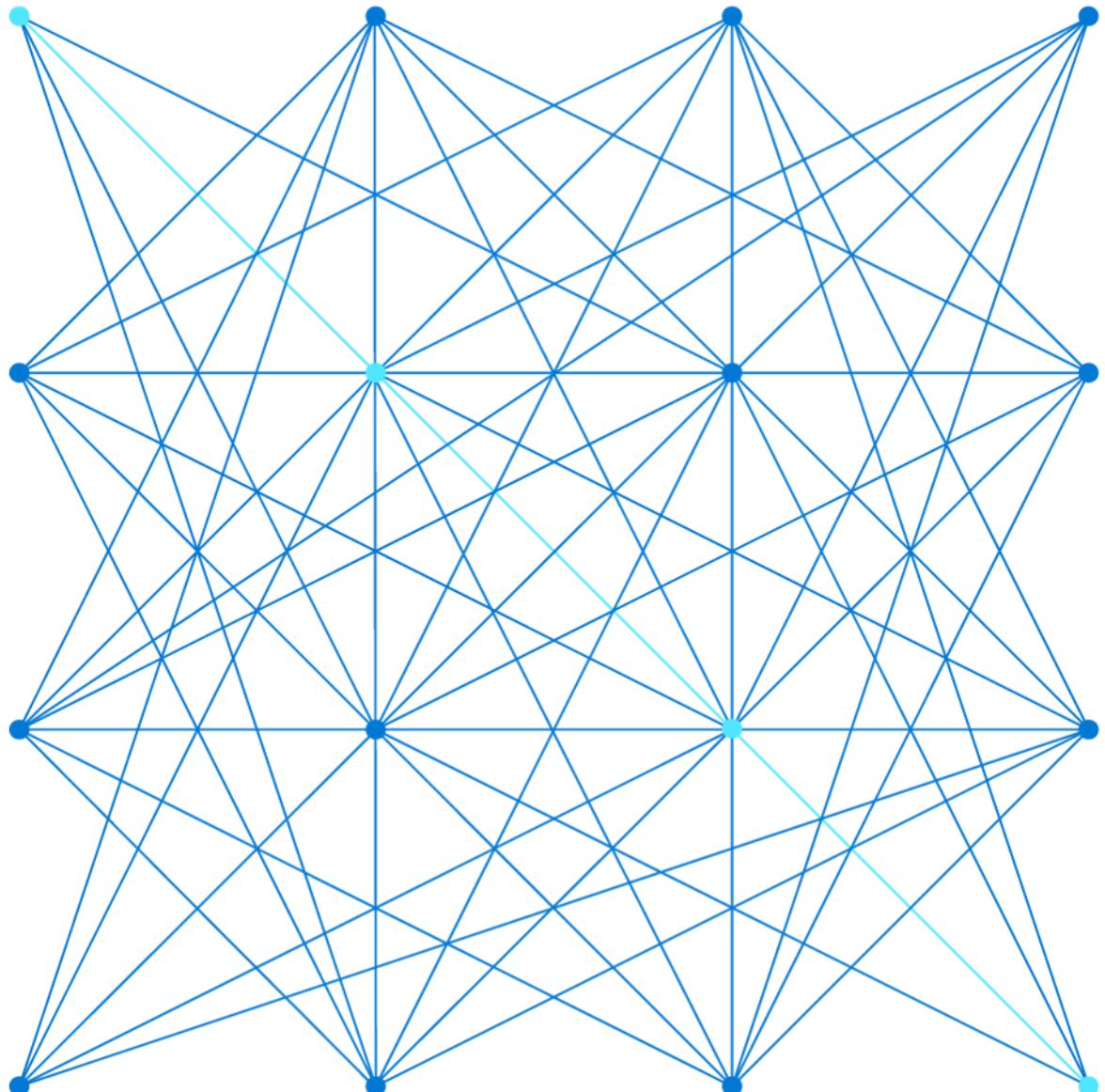
If the affected region is the current write region and automatic fail-over is enabled, then the region is automatically marked as offline. Then, an alternative region is promoted as the write region

## Lab: Building globally distributed databases with Cosmos DB

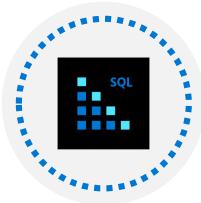




# Module 05: Working with relational data stores in the cloud

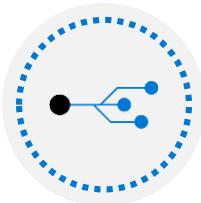


# Agenda



Lesson 01: Work with Azure SQL database

---



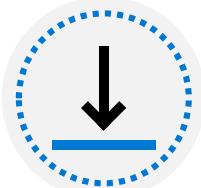
Lesson 02: Work with Azure Synapse Analytics

---



Lesson 03: Provision and query data in Azure Synapse Analytics

---



Lesson 04: Import data into Azure Synapse Analytics using PolyBase

# Lesson 01: Azure SQL database



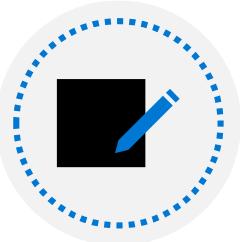
## Lesson objectives



Why Azure SQL Database is a good choice for running your relational database



What configuration and pricing options are available for your Azure SQL database

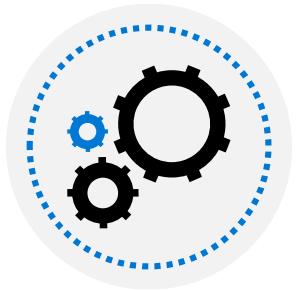


How to create an Azure SQL database from the portal



How to use Azure Cloud Shell to connect to your Azure SQL database, add a table, and work with data

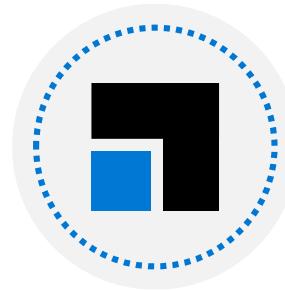
# Why Azure SQL Database is a good choice



Convenience



Cost



Scale



Security

# Azure SQL Database configuration options

When you create your first Azure SQL database, you also create an *Azure SQL logical server*. Think of a logical server as an administrative container for your databases

DTUs	vCores	SQL elastic pools	SQL managed instances
DTU stands for Database Transaction Unit and is a combined measure of compute, storage, and IO resources. Think of the DTU model as a simple, preconfigured purchase option	vCore gives you greater control over what compute and storage resources you create and pay for. vCore model enables you to configure resources independently	SQL elastic pools relate to eDTUs. They enable you to buy a set of compute and storage resources that are shared among all the databases in the pool. Each database can use the resources they need	The SQL managed instance creates a database with near 100% compatibility with the latest SQL Server on-premises Enterprise Edition database engine, useful for SQL Server customers who would like to migrate on-premises servers instance in a “lift and shift” manner

# Create an Azure SQL database

Home > New > SQL Database > Create SQL Database

## Create SQL Database

Microsoft

**Basics** • Networking Additional settings Tags Review + create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

**Subscription** \* ⓘ chtestao

**Resource group** \* ⓘ Select existing...  
Create new

**Database details**

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

**Database name** \* Enter database name

**Server** \* ⓘ Select a server  
Create new

The value must not be empty.

**Want to use SQL elastic pool?** \* ⓘ  Yes  No

**Compute + storage** \* ⓘ Please select a server first.  
[Configure database](#)

Dashboard > New > Create SQL Database

## 1 Create SQL Database

Microsoft

**2 Basics** Additional settings Tags Review + create

Customize additional configuration parameters including collation & sample data.

**Data source**

Start with a blank database, restore from a backup or select sample data to populate your new database.

\* Use existing data  None  Backup  Sample

\* Backup

3 Select a backup

myserver (West Europe)

database1 (2019-09-16 (12:05:30 UTC))
database2 (2019-09-16 (12:06:45 UTC))
database3 (2019-09-16 (12:07:51 UTC))
database4 (2019-09-16 (12:08:38 UTC))
database5 (2019-09-16 (12:09:23 UTC))
database6 (2019-09-16 (12:10:41 UTC))
database7 (2019-09-16 (12:11:38 UTC))

4 You can also restore a database to a server blade. [Learn more](#)

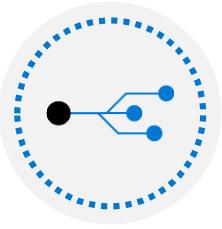
**Database Collation**

Database collation defines the rules that sort and compare data, and cannot be changed after database creation. The default database collation is SQL\_Latin1\_General\_CI\_AS. [Learn more](#)

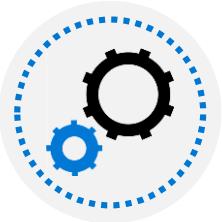
## Lesson 02: Azure Synapse Analytics



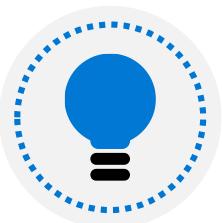
## Lesson objectives



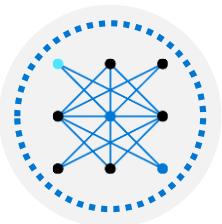
Explain Azure Synapse Analytics



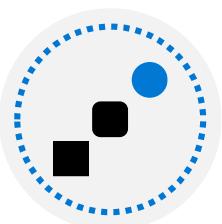
Explain Azure Synapse Analytics features



Types of solution workloads



Explain massively parallel processing concepts



Compare table geometries

# Azure Synapse Analytics

## What is Azure Synapse Analytics?

A unified environment by combining the enterprise data warehouse of SQL, the Big Data analytics capabilities of Spark, and data integration technologies to ease the movement of data between both, and from external data sources

## Data warehouse capabilities

### SQL Analytics:

A centralized data warehouse store that provides a relational analytics and decision support services across the whole enterprise

### SQL Pools:

CPU, memory, and IO are bundled into units of compute scale called SQL, determined by Data Warehousing Units (DWU)

### Future features:

Will include a Spark engine, a data integration and Azure Synapse Analytics Studio

# Azure Synapse Analytics features

## Workload management

This capability is used to prioritize the query workloads that take place on the server using Workload Management. This involves three components:

- Workload Groups
- Workload Classification
- Workload Importance

## Result-set cache

Result-set caching can be used to improve the performance of the queries that retrieve these results. When result-set caching is enabled, the results of the query are cached in the SQL pool storage

## Materialized views

A materialized view pre-computes, stores, and maintains its data like a table. They are automatically updated when data in underlying tables are changed

## SSDT CI/CD support

Database project support in SQL Server Data Tools (SSDT) allows teams of developers to collaborate over a version-controlled Azure Synapse Analytics, and track, deploy and test schema changes

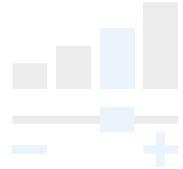
# Types of solution workloads

The modern data warehouse extends the scope of the data warehouse to serve Big Data that's prepared with techniques beyond relational ETL



## Modern data warehousing

We want to integrate all our data—including Big Data—with our data warehouse



## Advanced analytics

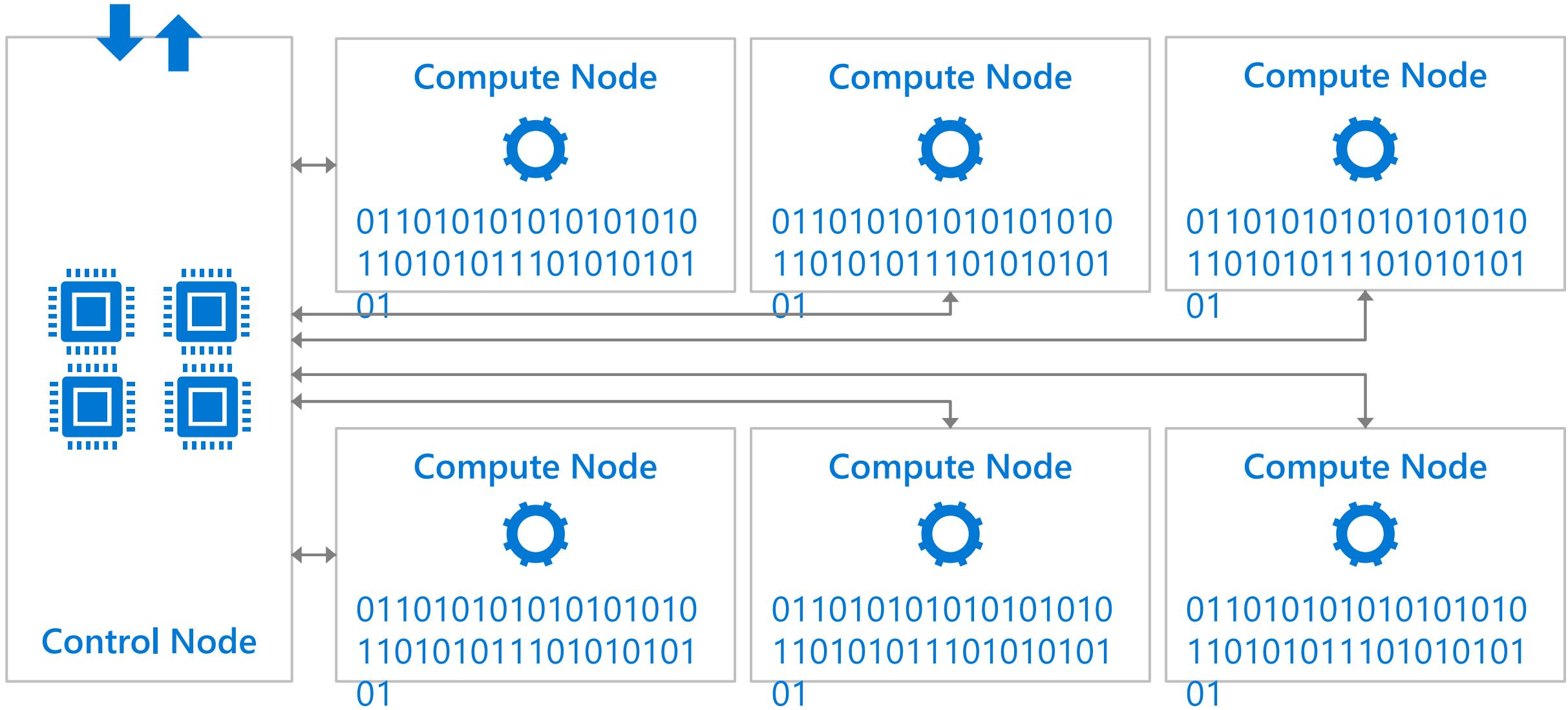
We're trying to predict when our customers churn



## Real-time analytics

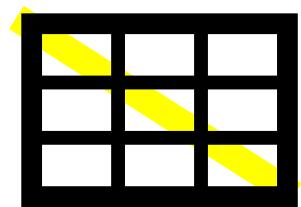
We're trying to get insights from our devices in real-time

# Massively Parallel Processing (MPP) concepts

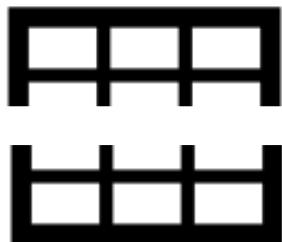


# Table geometries

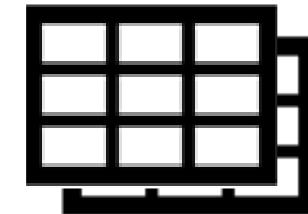
## Table distribution



Round Robin Tables

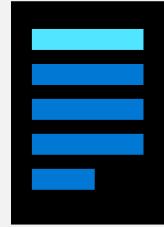


Hash Distributed Tables



Replicated Tables

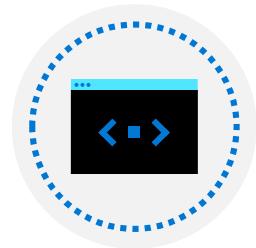
## Lesson 03: Creating and querying an Azure Synapse Analytics



## Lesson objectives



Create an Azure Synapse Analytics sample database



Query the sample database with the SELECT statement and its clauses



Use the queries in different client applications such as SQL Server Management Studio, and PowerBI

# Create an Azure Synapse Analytics

Home > New > Azure Synapse Analytics (formerly SQL DW) > SQL Data Warehouse

## SQL Data Warehouse

Welcome to Azure Synapse Analytics (formerly known as Azure SQL Data Warehouse). [Learn more](#)

**Basics** • Additional settings \* Tags Review + create

Create a SQL data warehouse with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ chtestao

Resource group \* ⓘ Select existing... Create new

**Data warehouse details**

Enter required settings for this data warehouse, including picking a logical server and configuring the performance level.

Data warehouse name \* Enter data warehouse name

Server \* ⓘ Select a server Create new

The value must not be empty.

Performance level \* ⓘ Please select a server first. Select performance level

# Perform Azure Synapse Analytics queries

## SELECT Query Basics

```
SELECT <select_list>
[FROM <optional_from_specification>]
[WHERE <optional_filter_condition>]
[ORDER BY <optional_sort_specification>]
[JOIN <optional_join_specification>]
```

## Examples

```
SELECT *
FROM Products p WHERE p.id ="1"
```

```
SELECT p.id, p.manufacturer, p.description
FROM Products p WHERE p.id ="1"
```

```
SELECT p.price, p.description, p.productId
FROM Products p ORDER BY p.price ASC
```

```
SELECT p.productId
FROM Products p JOIN p.shipping
```

# Perform Azure Synapse Analytics queries

## Create Table as Select (CTAS)

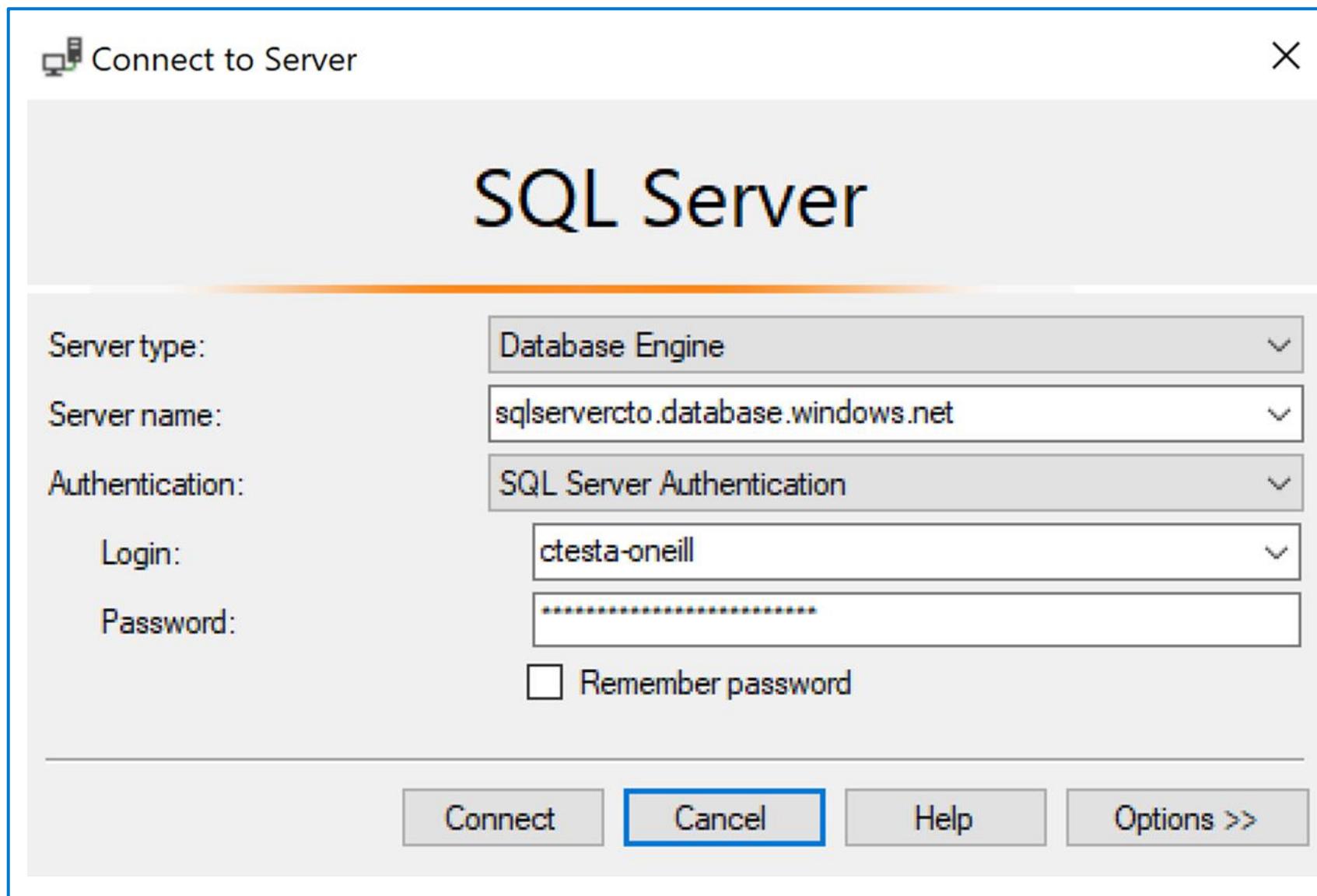
Used in parallel data loads

```
CREATE TABLE  
[ database_name . [ schema_name ] . | schema_name. ] table_name  
    [ ( { column_name } [ ,...n ] ) ]  
WITH ( DISTRIBUTION =  
    { HASH( distribution_column_name )           REPPLICATE | ROUND_ROBIN }  
        [ , <CTAS_table_option> [ ,...n ] ]  
    )  
AS <select_statement> [;]
```

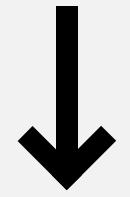
## Example

```
CREATE TABLE FactInternetSales_Copy  
WITH  
(DISTRIBUTION = HASH(SalesOrderNumber))  
AS SELECT * FROM FactInternetSales
```

# Querying with different client applications



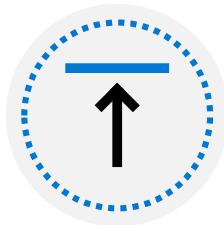
## Lesson 04: Using PolyBase to Load Data in Azure Synapse Analytics



## Lesson objectives



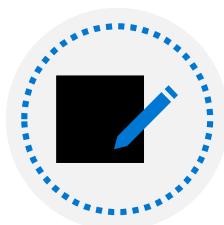
Explore how PolyBase works



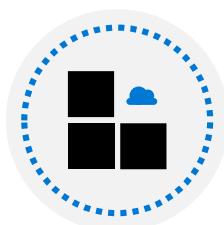
Upload text data to Azure Blob store



Collect the security keys for Azure Blob store



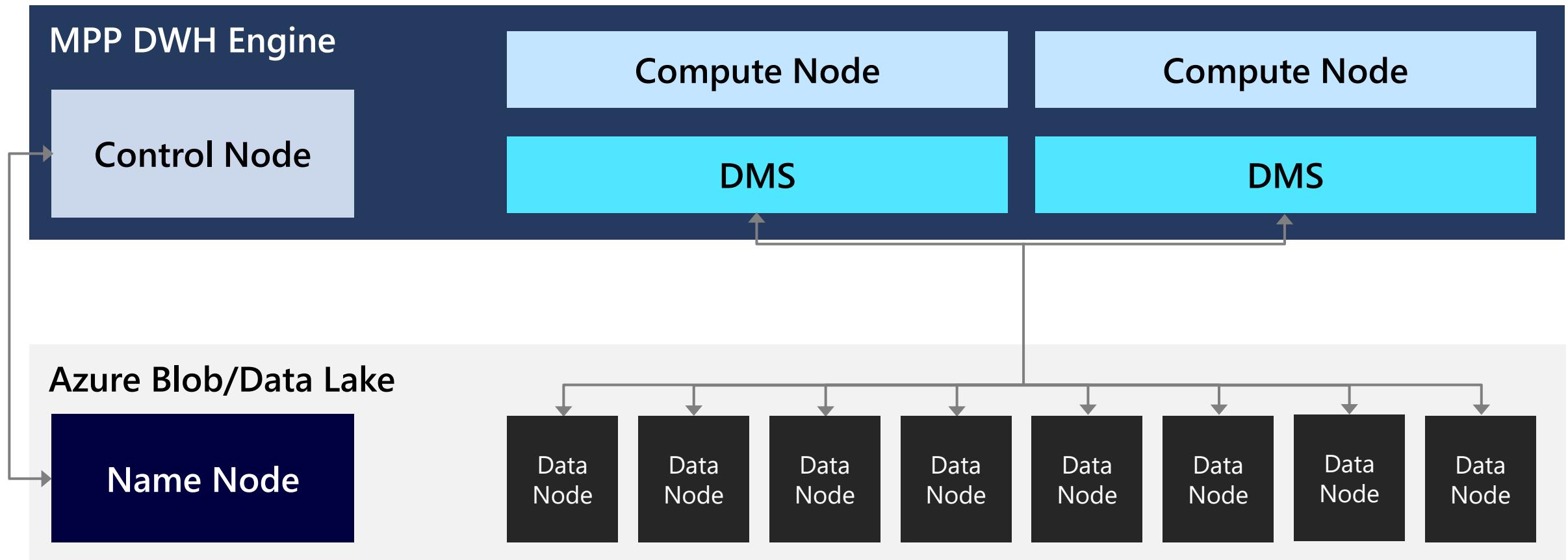
Create an Azure Synapse Analytics



Import data from Blob Storage to the Data Warehouse

# How PolyBase works

## The MPP engine's integration method with PolyBase



# Upload text data to Azure Blob store

Home > New > Storage account > Create storage account

## Create storage account

[Basics](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

**PROJECT DETAILS**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription: [dropdown menu]

\* Resource group: [dropdown menu] [Create new](#)

**INSTANCE DETAILS**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

\* Storage account name: [text input field]

\* Location: West Europe [dropdown menu]

Performance:  Standard  Premium

Account kind: StorageV2 (general purpose v2) [dropdown menu]

Replication: Read-access geo-redundant storage (RA-GRS) [dropdown menu]

Access tier (default):  Cool  Hot

[Review + create](#) [Previous](#) [Next : Advanced >](#)

# Collect the storage keys

toazureblob - Access keys

## Keys

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure K Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will interrupt access to disks from your virtual machines. [Learn more](#)

Storage account name  
ctoazureblob

**key1** 

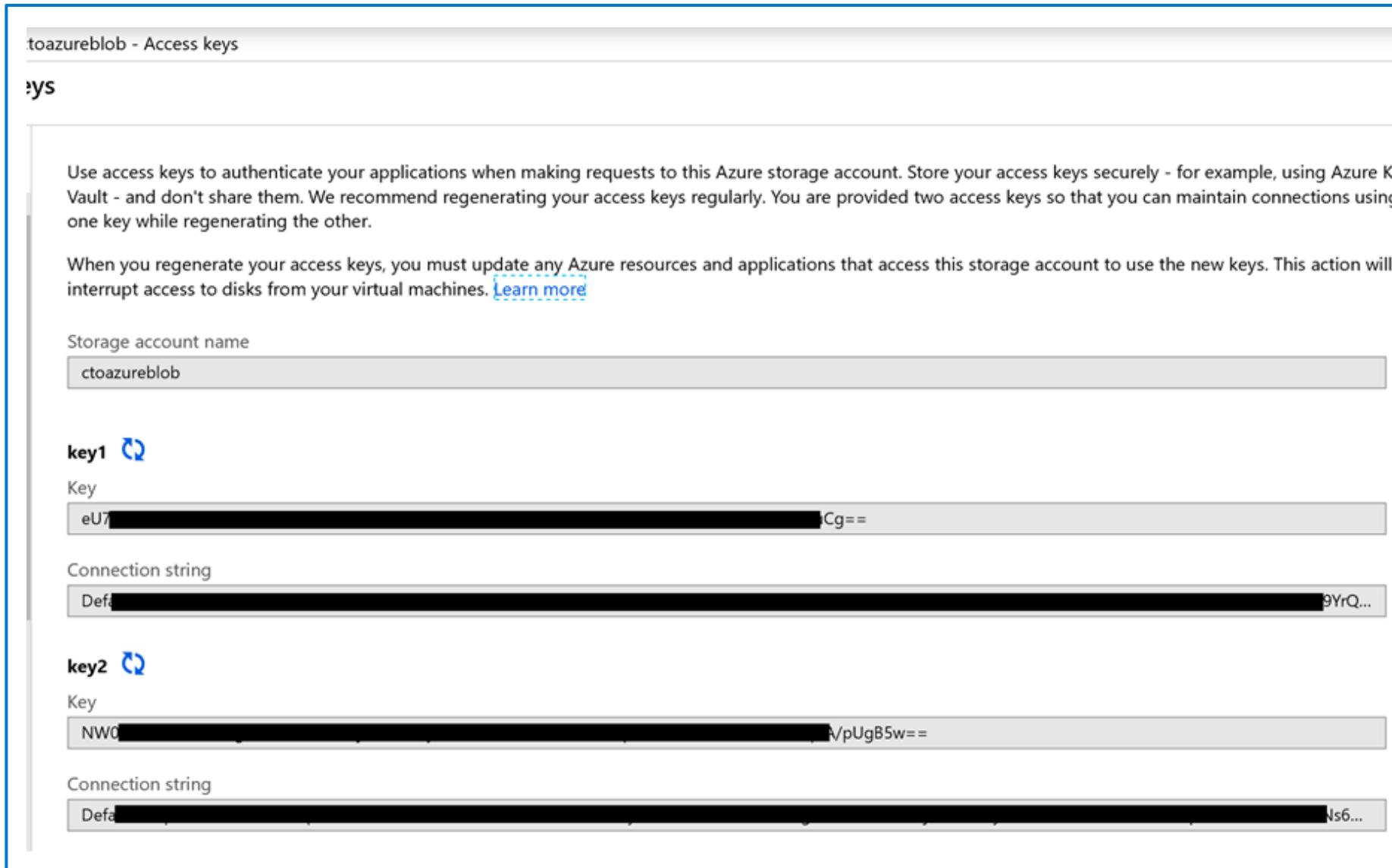
Key  
eU7...Cg==

Connection string  
Defa...9YrQ...

**key2** 

Key  
NWO...V/pUgB5w==

Connection string  
Defa...Ns6...



# Create an Azure Synapse Analytics

The screenshot shows the Azure portal interface for creating a SQL Data Warehouse. The top navigation bar includes 'Home', 'New', 'Azure Synapse Analytics (formerly SQL DW)', and 'SQL Data Warehouse'. The main title is 'SQL Data Warehouse' by Microsoft. A purple banner at the top says 'Welcome to Azure Synapse Analytics (formerly known as Azure SQL Data Warehouse)'. Below the banner, there are tabs: 'Basics' (underlined), 'Additional settings\*', 'Tags', and 'Review + create'. A note below the tabs says: 'Create a SQL data warehouse with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize.' A 'Learn more' link is provided. The 'Project details' section asks to select a subscription and resource group. The 'Subscription' dropdown is set to 'chtestao'. The 'Resource group' dropdown is set to 'Select existing...' with a 'Create new' link below it. The 'Data warehouse details' section requires entering a 'Data warehouse name' and selecting a 'Server'. The 'Data warehouse name' field is empty. The 'Server' dropdown is empty and highlighted with a red border, with a red error message: 'The value must not be empty.' The 'Performance level' section is also visible but empty.

Home > New > Azure Synapse Analytics (formerly SQL DW) > SQL Data Warehouse

## SQL Data Warehouse

Welcome to Azure Synapse Analytics (formerly known as Azure SQL Data Warehouse). [Learn more](#)

**Basics** • Additional settings\* Tags Review + create

Create a SQL data warehouse with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ chtestao

Resource group \* ⓘ Select existing... [Create new](#)

### Data warehouse details

Enter required settings for this data warehouse, including picking a logical server and configuring the performance level.

Data warehouse name \* Enter data warehouse name

Server \* ⓘ Select a server [Create new](#)

✖ The value must not be empty.

Performance level \* ⓘ Please select a server first. [Select performance level](#)

## Lab: Working with relational data stores in the cloud



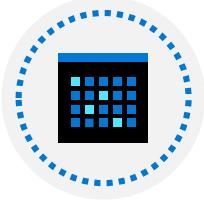


# Module 06:

## Performing real-time analytics with Stream Analytics

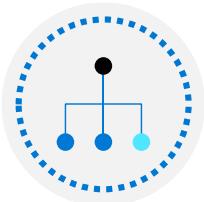


# Agenda



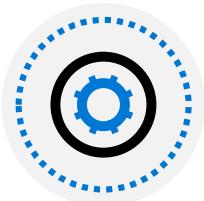
Lesson 01 – Data streams and event processing

---



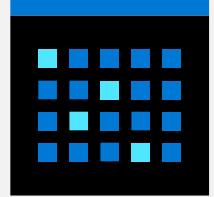
Lesson 02 – Data ingestion with Event Hubs

---

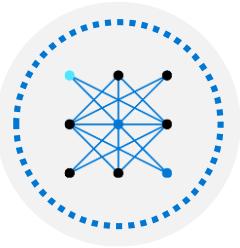


Lesson 03 – Processing data with Stream Analytics Jobs

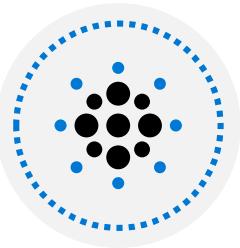
# Lesson 01: Data streams and event processing



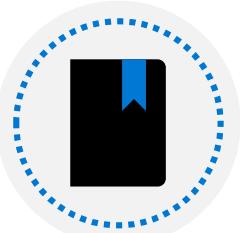
## Lesson objectives



Explain data streams



Explain event processing



Learn about processing events with Azure Stream Analytics

# What are data streams

## Data streams:

In the context of analytics, data streams are event data generated by sensors or other sources that can be analyzed by another technology

## Data stream processing approach:

There are two approaches. Reference data is streaming data that can be collected over time and persisted in storage as static data. In contrast, streaming data have relatively low storage requirements. And run computations in sliding windows

## Data streams are used to:

### Analyze data:

Continuously analyze data to detect issues and understand or respond to them

### Understand systems:

Understand component or system behavior under various conditions to fuel further enhancements of said system

### Trigger actions:

Trigger specific actions when certain thresholds are identified

# Event processing

The process of consuming data streams, analyzing them, and deriving actionable insights out of them is called Event Processing and has three distinct components:

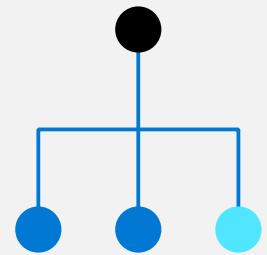
Event producer	Examples include sensors or processes that generate data continuously such as a heart rate monitor or a highway toll lane sensor
Event processor	An engine to consume event data streams and deriving insights from them. Depending on the problem space, event processors either process one incoming event at a time (such as a heart rate monitor) or process multiple events at a time (such as a highway toll lane sensor)
Event consumer	An application which consumes the data and takes specific action based on the insights. Examples of event consumers include alert generation, dashboards, or even sending data to another event processing engine

# Processing events with Azure Stream Analytics

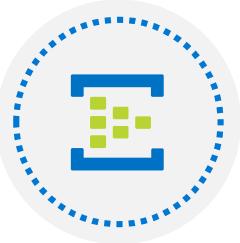
Microsoft Azure Stream Analytics is an event processing engine. It enables the consumption and analysis of high volumes of streaming data in real time

Source	Ingestion	Analytical engine	Destination
Sensors	Event Hubs	Stream Analytics Query Language .NET SDK	Azure Data Lake
Systems	IoT Hubs		Cosmos DB
Applications	Azure Blob Store		SQL Database Blob Store Power BI

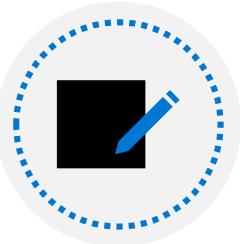
## Lesson 02: Data ingestion with Event Hubs



## Lesson objectives



Describe Azure Event Hubs



Create an Event Hub



Evaluate the performance of an Event Hub

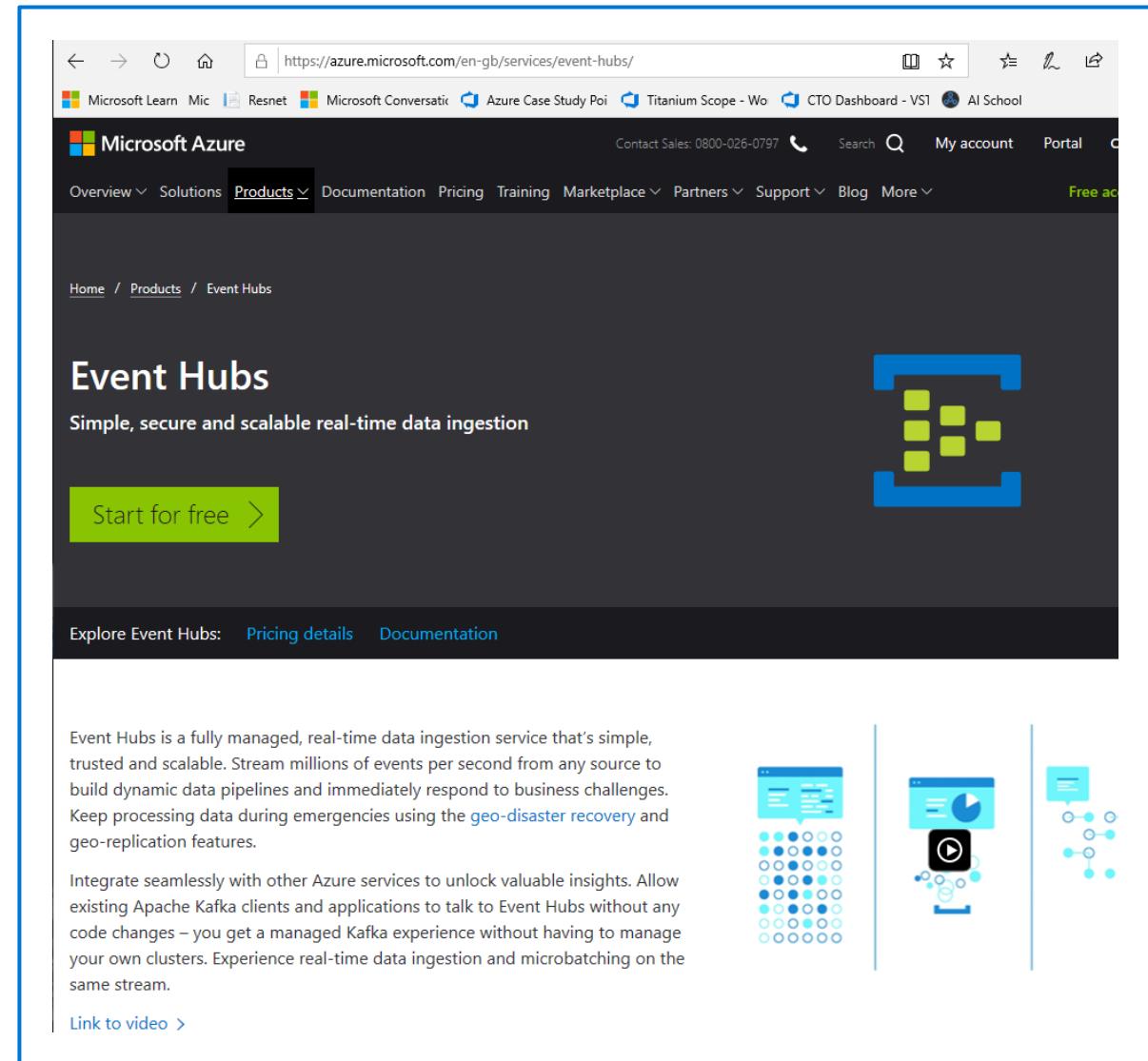


Configure applications to use an Event Hub

# Azure Event Hubs



*Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple applications*



The screenshot shows the Microsoft Azure website at https://azure.microsoft.com/en-gb/services/event-hubs/. The page title is "Event Hubs". The main heading is "Event Hubs" with the subtext "Simple, secure and scalable real-time data ingestion". A green button says "Start for free >". Below the heading, there's a description of Event Hubs as a fully managed, real-time data ingestion service. To the right, there are three icons representing different Azure services: a database, a chart, and a network diagram.

https://azure.microsoft.com/en-gb/services/event-hubs/

Microsoft Azure

Overview Solutions Products Documentation Pricing Training Marketplace Partners Support Blog More

Contact Sales: 0800-026-0797 Search My account Portal

Home / Products / Event Hubs

## Event Hubs

Simple, secure and scalable real-time data ingestion

Start for free >

Explore Event Hubs: Pricing details Documentation

Event Hubs is a fully managed, real-time data ingestion service that's simple, trusted and scalable. Stream millions of events per second from any source to build dynamic data pipelines and immediately respond to business challenges. Keep processing data during emergencies using the geo-disaster recovery and geo-replication features.

Integrate seamlessly with other Azure services to unlock valuable insights. Allow existing Apache Kafka clients and applications to talk to Event Hubs without any code changes – you get a managed Kafka experience without having to manage your own clusters. Experience real-time data ingestion and microbatching on the same stream.

[Link to video >](#)

# Create an Event Hub

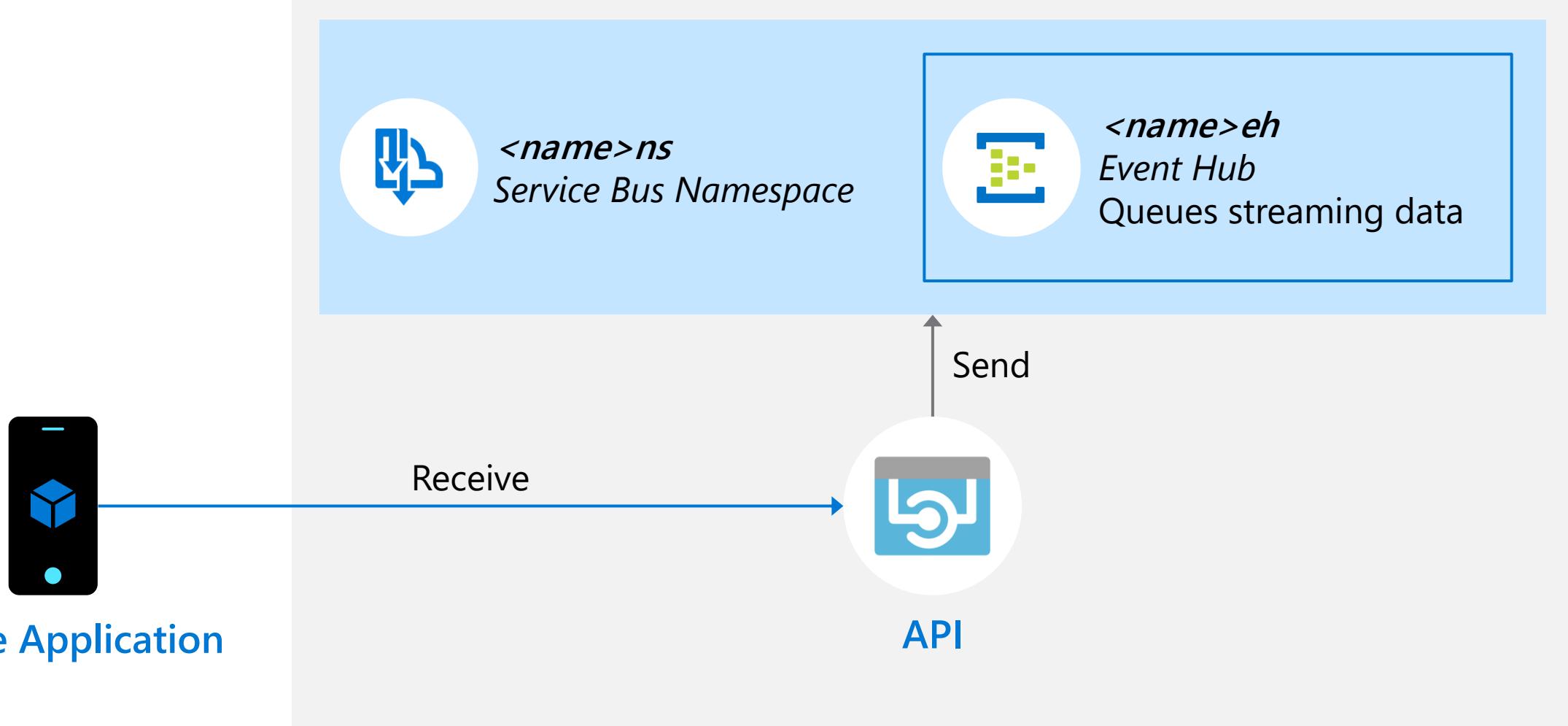
## Create an event hub namespace

1. In the [Azure portal](#), select NEW, type Event Hubs, and then select Event Hubs from the resulting search. Then select Create
2. Provide a name for the event hub, and then create a resource group. Specify **xx-name-eh** and **xx-name-rg** respectively, XX- represent your initials to ensure uniqueness of the Event Hub name and Resource Group name
3. Click the checkbox to **Pin to the dashboard**, then select the **Create** button

## Create an event hub

1. After the deployment is complete, click the **xx-name-eh** event hub on the dashboard
2. Then, under **Entities**, select **Event Hubs**
3. To create the event hub, select the **+ Event Hub** button. Provide the name **socialstudy-eh**, and then select **Create**
4. To grant access to the event hub, we need to create a shared access policy. Select the **socialstudy-eh** event hub when it appears, and then, under **Settings**, select **Shared access policies**
5. Under **Shared access policies**, create a policy with **MANAGE** permissions by selecting **+ Add**. Give the policy the name of **xx-name-eh-sap**, check **MANAGE**, and then select **Create**
6. Select your new policy after it has been created, and then select the copy button for the **CONNECTION STRING – PRIMARY KEY** entity
7. Paste the **CONNECTION STRING – PRIMARY KEY** entity into Notepad, this is needed later in the exercise
8. Leave all windows open

# Configure applications to use Event Hubs



# Evaluating the performance of Event Hubs

Search (Ctrl+ /) < Consumer group Delete

**Overview** (highlighted with a red box)

Resource group (change)  
ehtestrg

Status Active

Location UK West

Subscription (change)  
Free Trial

Subscription ID d73c53de-f991-4b7d-ad7b-b2a01...

Namespace ehntest78

Created Monday, 16 July 2018

Updated Monday, 16 July 2018

EVENT HUB CONTENTS 1 CONSUMER GROUP

EVENT HUB STATUS ACTIVE

MESSAGE RETENTION 7 DAYS

Show metrics data for the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

**Requests**

110  
100  
90  
80  
70  
60  
50  
40  
30  
20  
10  
0

10:45 11:00 11:15 11:30

INCOMING REQUESTS... EHNTTEST78 121

SUCCESSFUL REQUESTS... EHNTTEST78 121

SERVER ERRORS... EHNTTEST78 --

**Messages**

110  
100  
90  
80  
70  
60  
50  
40  
30  
20  
10  
0

10:45 11:00 11:15 11:30

INCOMING MESSAGES... EHNTTEST78 100

OUTGOING MESSAGES... EHNTTEST78 100

CAPTURED MESSAGES... EHNTTEST78 --

**Throughput**

1.8kB  
1.6kB  
1.4kB  
1.2kB  
1kB  
800B  
600B  
400B  
200B  
0B

10:45 11:00 11:15 11:30

INCOMING BYTES... EHNTTEST78 1.69kB

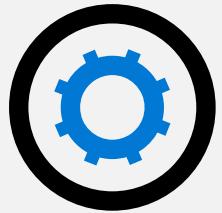
OUTGOING BYTES... EHNTTEST78 1.69kB

CAPTURED BYTES... EHNTTEST78 --

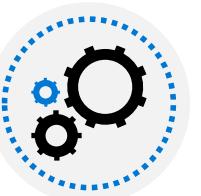
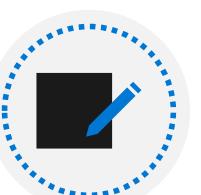
NAME \$Default

LOCATION UK West

## Lesson 03: Processing data with Stream Analytics Jobs

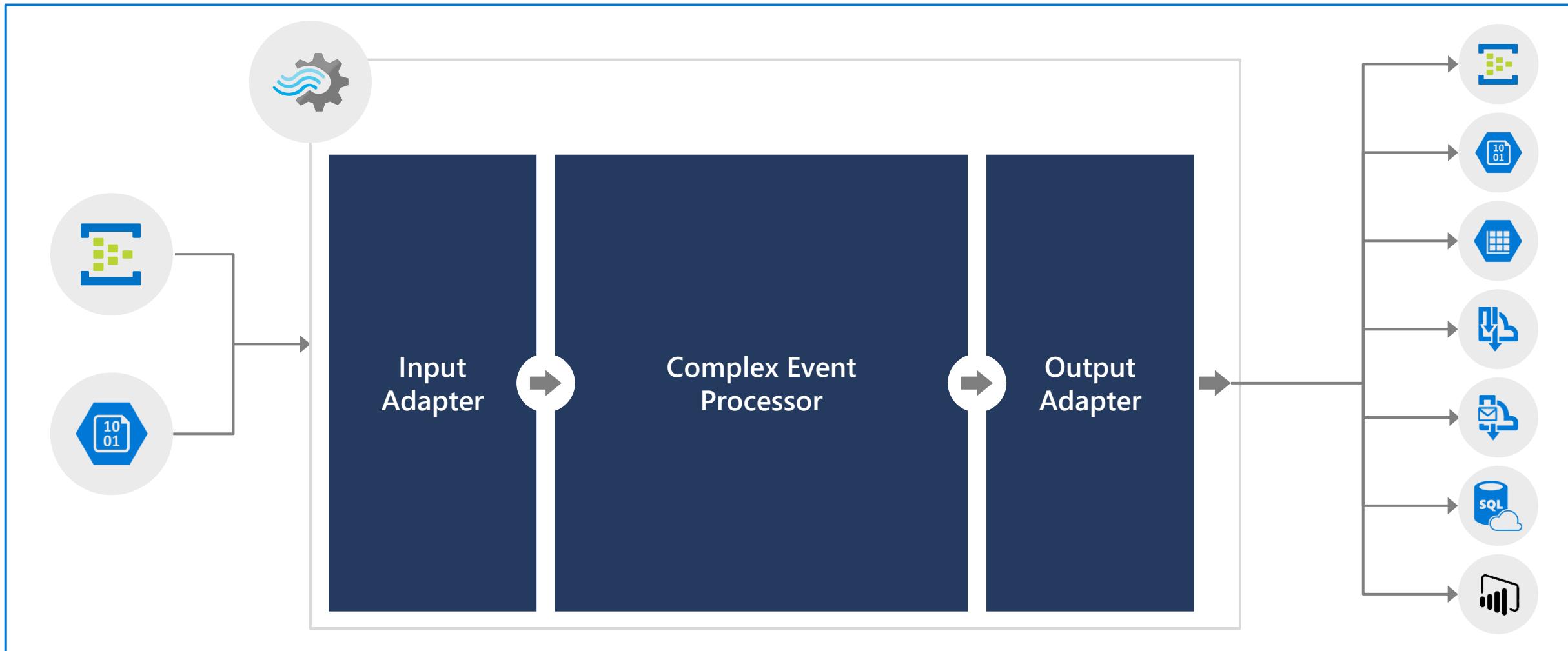


## Lesson objectives

-  Explore the Streaming Analytics workflow
-  Create a Stream Analytics Job
-  Configure a Stream Analytics job input
-  Configure a Stream Analytics job output
-  Write a transformation query
-  Start a Stream Analytics job

# Azure Stream Analytics workflow

Complex event processing of Stream Data in Azure



# Create Stream Analytics service

Job name

Subscription

Resource group

Location

Home > New > Stream Analytics job > New Stream Analytics job

New Stream Analytics job

\* Job name  
cto-asa-job1

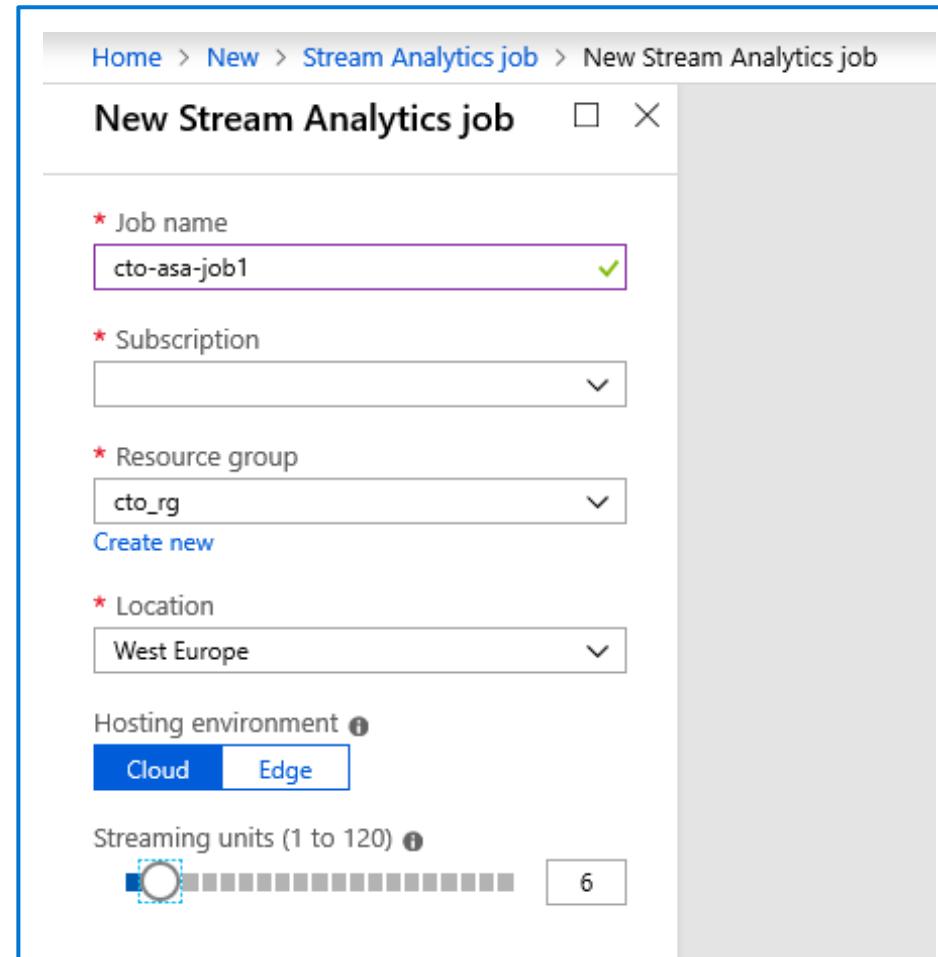
\* Subscription

\* Resource group  
cto\_rg   
[Create new](#)

\* Location  
West Europe

Hosting environment

Streaming units (1 to 120)



# Create a Stream Analytics Job input

**Event Hub** >

New input

\* Input alias  
cto-asa-input01 ✓

Provide Event Hub settings manually  
 Select Event Hub from your subscriptions

Subscription  
LearnAI Training Subscription ▾

\* Event Hub namespace ⓘ  
cto-eh-ns ▾

\* Event Hub name ⓘ  
 Create new  Use existing  
cto-name-eh ▾

\* Event Hub policy name ⓘ  
RootManageSharedAccessKey ▾

Event Hub policy key  
\*\*\*\*\*

Event Hub consumer group ⓘ

\* Event serialization format ⓘ  
JSON ▾

Encoding ⓘ  
UTF-8 ▾

Event compression type ⓘ  
None ▾

# Create a Stream Analytics Job output

Home > Resource groups > cto\_rg > cto-asa-job1 > Outputs

## Outputs

**SINK**

**Add**

- Event Hub
- SQL Database
- Blob storage**
- Table storage
- Service Bus topic
- Service Bus queue
- Cosmos DB
- Power BI
- Data Lake Storage Gen1

**Blob storage**  
New output

\* Output alias: cto-asa-output01 ✓

Provide Blob storage settings manually  
 Select Blob storage from your subscriptions

Subscription: LearnAI Training Subscription

\* Storage account: ctoazureblob

\* Storage account key: [REDACTED]

\* Container:  
 Create new  
 Use existing  
socialmedia ✓

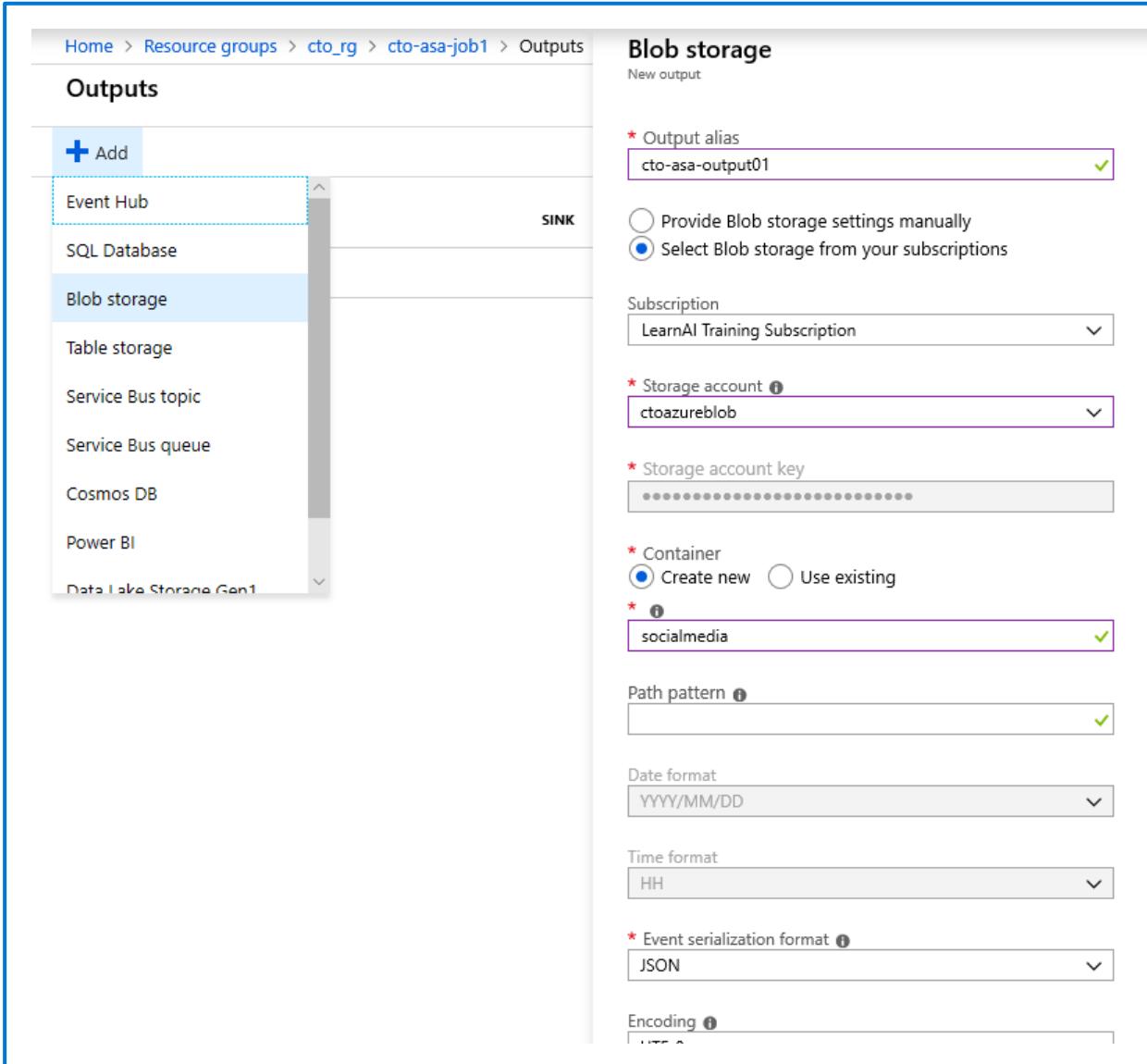
Path pattern: [REDACTED] ✓

Date format: YYYY/MM/DD

Time format: HH

\* Event serialization format: JSON

Encoding: [REDACTED]



# Write a transformation query

The screenshot shows the Azure Stream Analytics job overview page for 'cto-asa-job1'. The left sidebar contains navigation links: Overview (selected), Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Locks, Job topology, Inputs, Functions, Query, and Outputs. The main area displays job details: Resource group (change) : cto\_rg, Status : Created, Location : West Europe, Subscription (change) : LearnAI Training Subscription, and Subscription ID : 5be49961-ea44-42ec-8021-b728be90d58c. Below this are sections for Inputs (1 cto-asa-input01) and Outputs (1 cto-asa-output01). On the right, a red box highlights the 'Query' section which contains the following T-SQL code:

```
1 SELECT
2   *
3 INTO
4   [cto-asa-output01]
5 FROM
6   [cto-asa-input01]
```

# Start a Stream Analytics Job

The screenshot shows the Azure Stream Analytics job overview page for 'cto-asa-job1'. The job is currently in a 'Created' state. The 'Start' button is highlighted with a red box. The 'Inputs' section shows one input named 'cto-asa-input01'. The 'Outputs' section shows one output named 'cto-asa-output01'. The 'Query' section displays the following T-SQL code:

```
1 SELECT
2   *
3 INTO
4   [cto-asa-output01]
5 FROM
6   [cto-asa-input01]
```

The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Locks, Inputs, Functions, Query, Outputs, Configure, and Storage account settings.

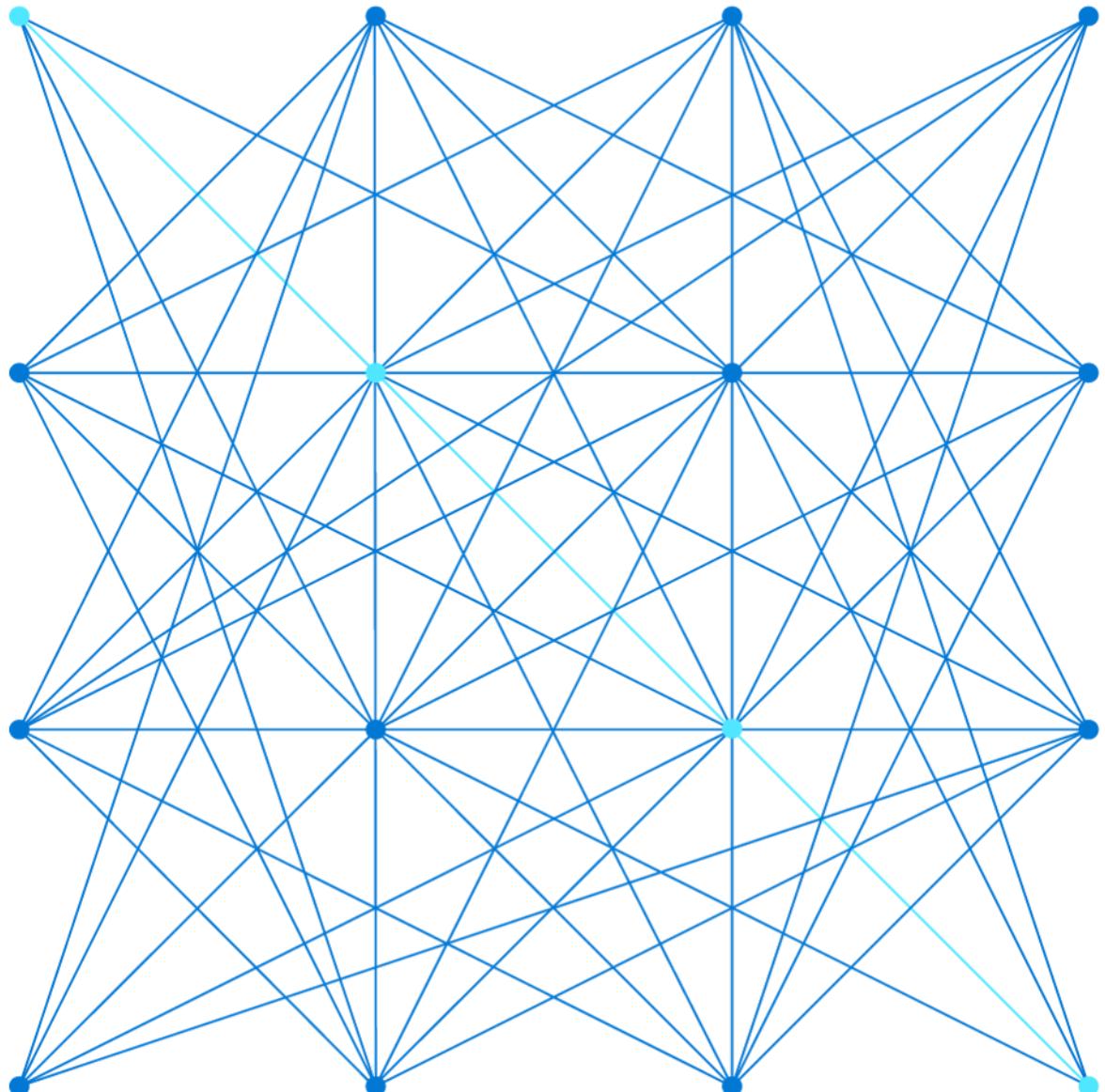
# Lab: Performing real-time analytics with Stream Analytics





# Module 07: Orchestrating data movement with Azure Data Factory

Start : 09.15



# Agenda



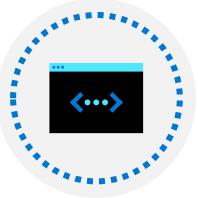
Lesson 01 – Introduction to Azure Data Factory

---



Lesson 02 – Understand Azure Data Factory components

---



Lesson 03 – Integrate Azure Data Factory with Databricks

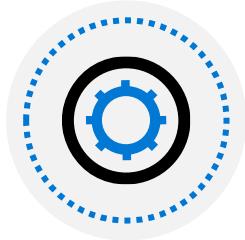
# Lesson 01: Introduction to Azure Data Factory



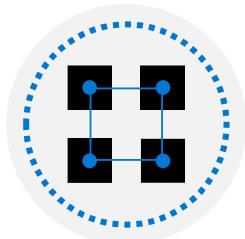
## Lesson objectives



What is Azure Data Factory



The Data Factory process

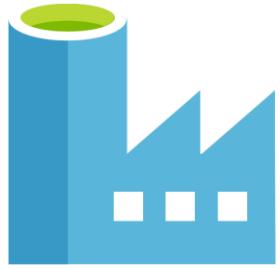


Azure Data Factory components



Azure Data Factory security

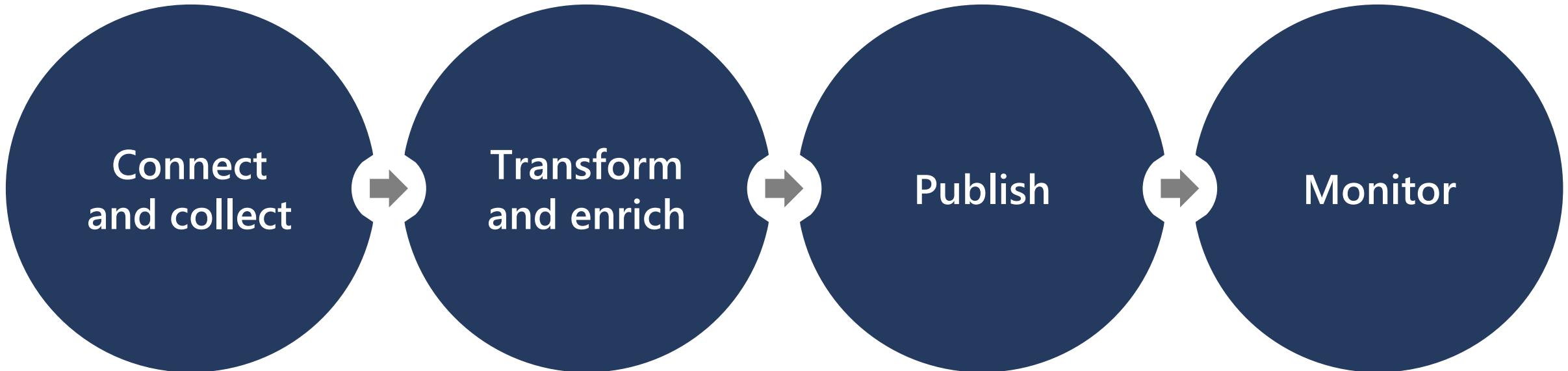
# What is Azure Data Factory



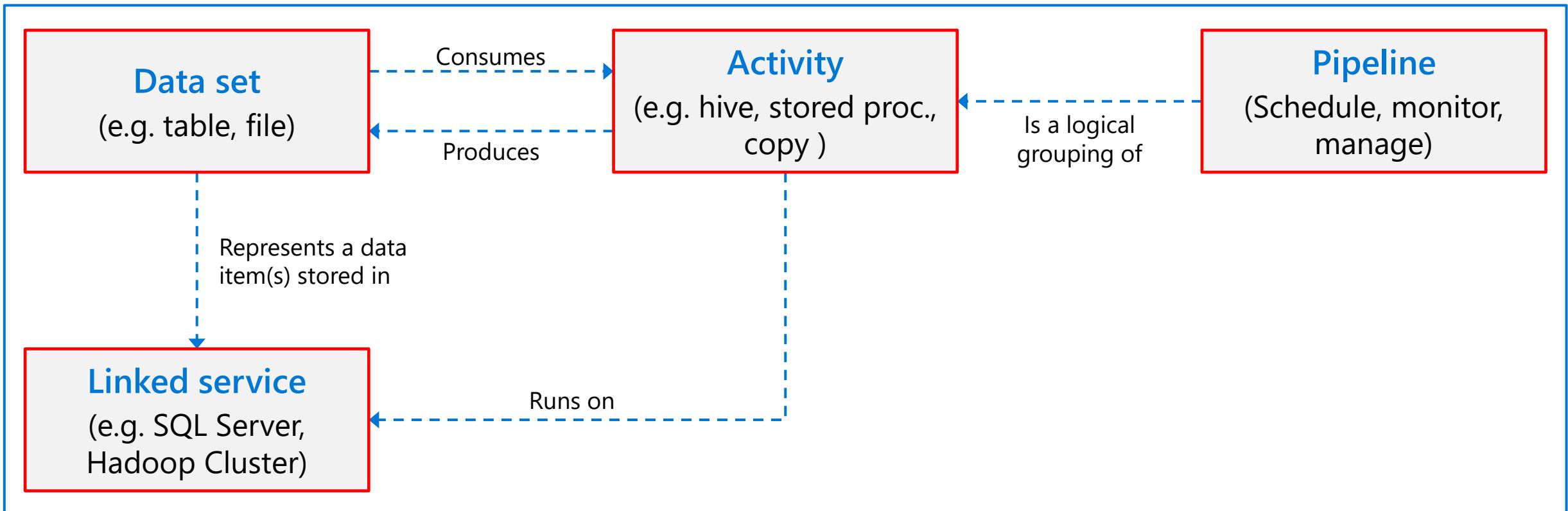
Creates, orchestrates, and automates the movement, transformation and/or analysis of data through in the cloud



# The Data Factory process



# Azure Data Factory components



Control flow

Parameters

Integration runtime

# Azure Data Factory security

## Data factory contributor role

1

Create, edit, and delete data factories and child resources including datasets, linked services, pipelines, triggers, and integration runtimes

2

Deploy Resource Manager templates. Resource Manager deployment is the deployment method used by Data Factory in the Azure portal

3

Manage App Insights alerts for a data factory

4

At the resource group level or above, lets users deploy Resource Manager template

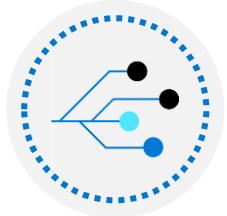
5

Create support tickets

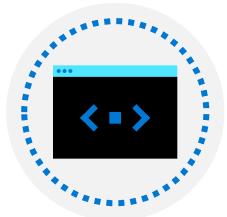
## Lesson 02: Azure Data Factory components



## Lesson objectives



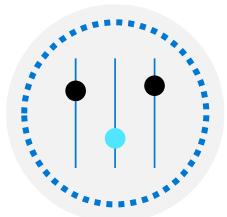
Linked services



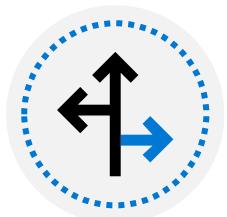
Datasets



Data Factory activities

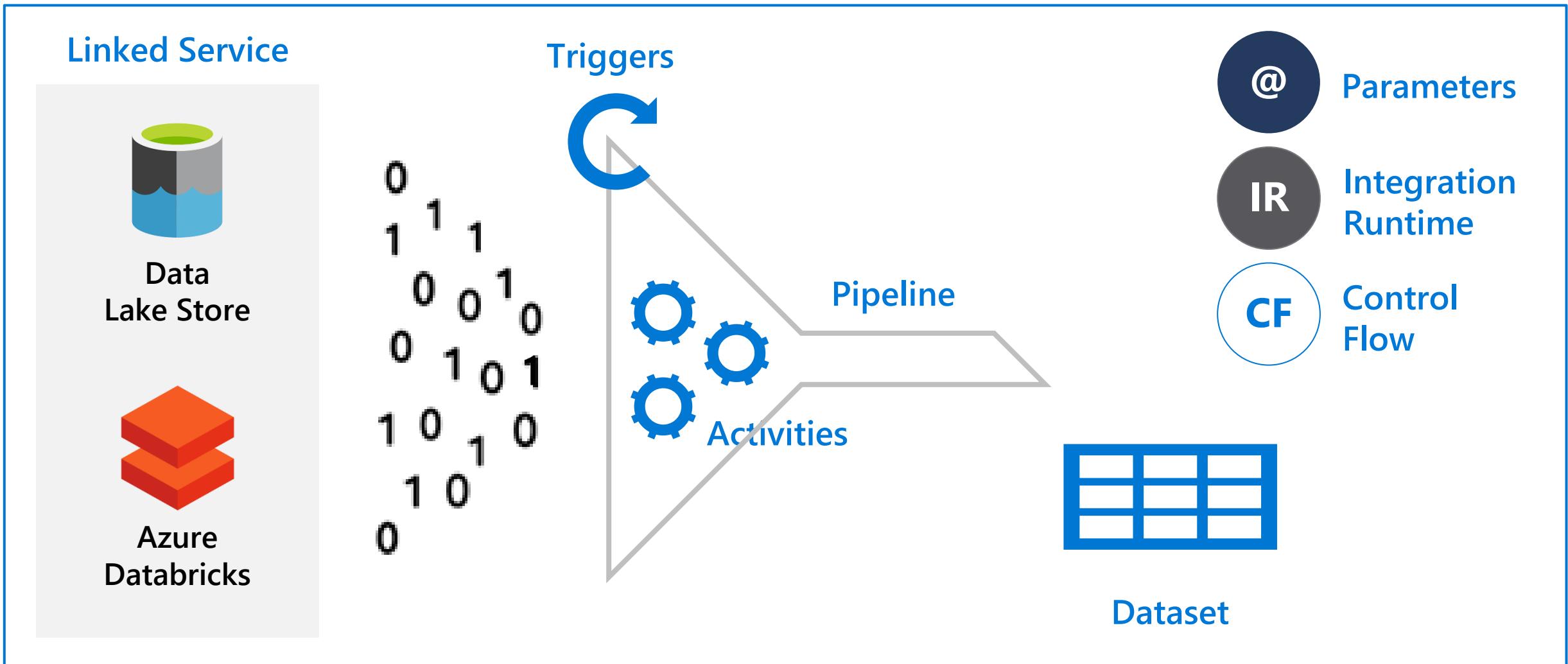


Pipelines



Pipeline example

# Azure Data Factory components



# Data factory activities

Activities within Azure Data Factory defines the actions that will be performed on the data and there are three categories including:

## 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

# Pipelines

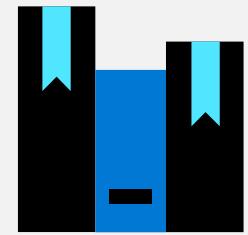


**Pipeline** is a grouping of logically related activities

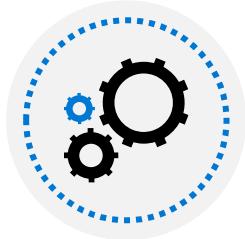
**Pipeline** can be scheduled so the activities within it get **executed**

**Pipeline** can be managed and monitored

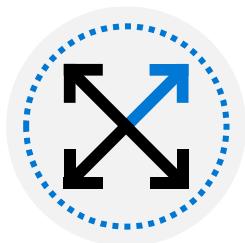
# Lesson 03: Ingesting and transforming data



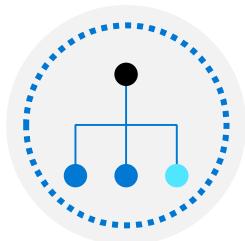
## Lesson objectives



How to setup Azure Data Factory



Ingest data using the Copy Activity



Transforming data with the Mapping Data Flow

# Create Azure Data Factory

Home > New > Data Factory > New data factory

### New data factory

Name \*

Version ⓘ

Subscription \*

Resource Group \*

Select existing...

Create new

Location \* ⓘ

Enable GIT ⓘ

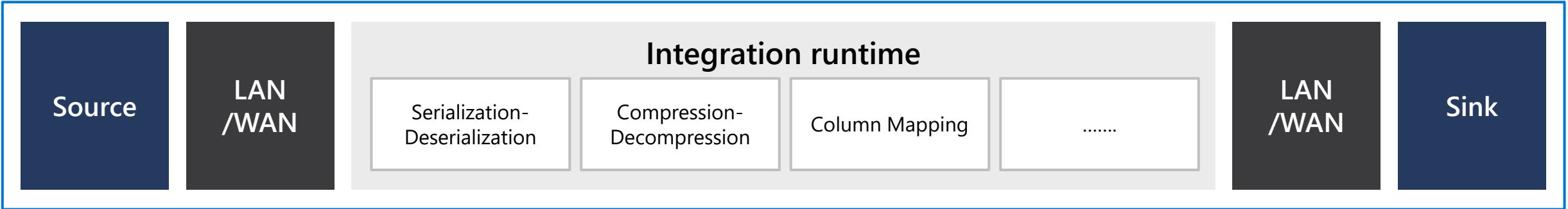
GIT URL \* ⓘ

Repo name \* ⓘ

Branch Name \* ⓘ

Root folder \* ⓘ

# Ingesting data with the copy activity



Reads data from a source data store

Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration of the input dataset, output dataset, and Copy activity

Writes data to the sink/destination data store

# Transforming data with the Mapping Data Flow

## Code free data transformation at scale

Perform data cleansing, transformation, aggregations, etc.

Enables you to build resilient data flows in a code free environment

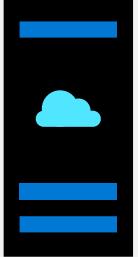
Enable you to focus on building business logic and data transformation

Underlying infrastructure is provisioned automatically with cloud scale via Spark execution

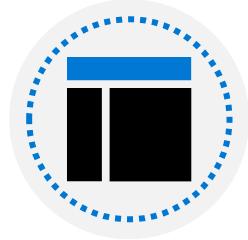
### Mapping Data Flow



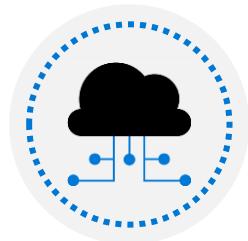
## Lesson 04: Integrate Azure Data Factory with Azure Databricks



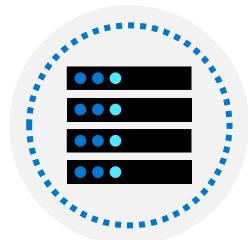
## Lesson objectives



Use Azure Data Factory (ADF) to ingest data and create an ADF pipeline



Create Azure Storage account and the Azure Data Factory instance



Use ADF to orchestrate data transformations using a Databricks Notebook activity

# Working with documents programmatically

1.  
Create Storage  
Account

2.  
Create Azure  
Data Factory

3.  
Create data  
workflow  
pipeline

4.  
Add Databricks  
Workbook to  
pipeline

5.  
Perform analysis  
on the data

# Create Azure Storage account and the Azure Data Factory instance

Home > New > Storage account > Create storage account

## Create storage account

[Basics](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

**PROJECT DETAILS**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription:

\* Resource group:  [Create new](#)

**INSTANCE DETAILS**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

\* Storage account name:

\* Location:  West Europe

Performance:  Standard  Premium

Account kind:  StorageV2 (general purpose v2)

Replication:  Read-access geo-redundant storage (RA-GRS)

Access tier (default):  Cool  Hot

[Review + create](#) [Previous](#) [Next : Advanced >](#)

Home > New > Data Factory > New data factory

## New data factory

Name \*:

Version:  V2

Subscription:  chtestao

Resource Group \*:  Select existing... [Create new](#)

Location \*:  South Central US

Enable GIT:

GIT URL \*:

Repo name \*:

Branch Name \*:

Root folder \*:

[Create](#)

# Use ADF to orchestrate data transformations using a Databricks Notebook activity

Microsoft Azure

03-Data-Transformation (Python)

Azure Databricks Home Workspace Recents Data Clusters Jobs Search

Detached File View: Code Permissions Run All Clear Schedule

Cmd 1

## Data Transformation via Azure Data Factory

As you saw at the end of the previous lesson, different cities use different field names and values to indicate crimes, dates, etc. within their crime data.

For example:

- Some cities use the value "HOMICIDE", "CRIMINAL HOMICIDE" or "MURDER".
- In the New York data, the column is named `offenseDescription` while in the Boston data, the column is named `OFFENSE_CODE_GROUP`.
- In the New York data, the date of the event is in the `reportDate`, while in the Boston data, there is a single column named `MONTH`.

In the case of New York and Boston, here are the unique characteristics of each data set:

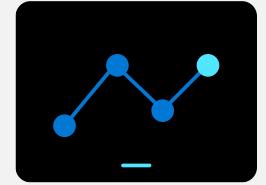
	Offense-Column	Offense-Value	Reported-Column	Reported-Data Type
New York	<code>offenseDescription</code>	starts with "murder" or "homicide"	<code>reportDate</code>	<code>timestamp</code>
Boston	<code>OFFENSE_CODE_GROUP</code>	"Homicide"	<code>MONTH</code>	<code>integer</code>

In this notebook, we will use an ADF Databricks Notebooks activity to perform transformations on and extract homicide statistics from the crime data being

In this lesson you:

1. Create Databricks Access Token.
2. Add Databricks Notebook activity to pipeline.
3. Connect Copy Activities to Notebook Activity.
4. Publish the updated pipeline.
5. Trigger and Monitor the pipeline run.
6. Verify transformations of data by looking at the generated table in Databricks.
7. Perform a simple aggregation of the data.

# Lab: Orchestrating data movement with Azure Data Factory





# Module 08:

# Securing Azure

# Data Platforms



# Agenda



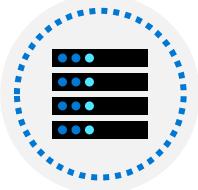
Lesson 01 – An introduction to security

---



Lesson 02 – Key security components

---



Lesson 03 – Securing storage accounts and Data Lake Storage

---



Lesson 04 – Securing data stores

---

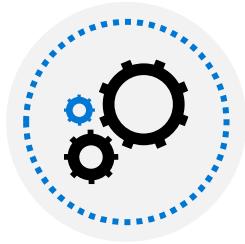


Lesson 05 – Securing streaming data

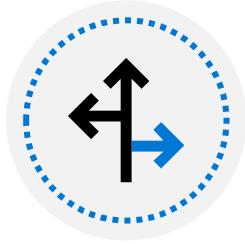
# Lesson 01: An Introduction to security



## Lesson objectives



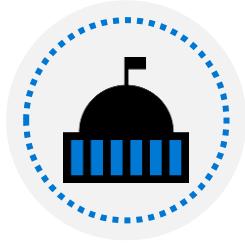
Shared security responsibility



A layered approach to security

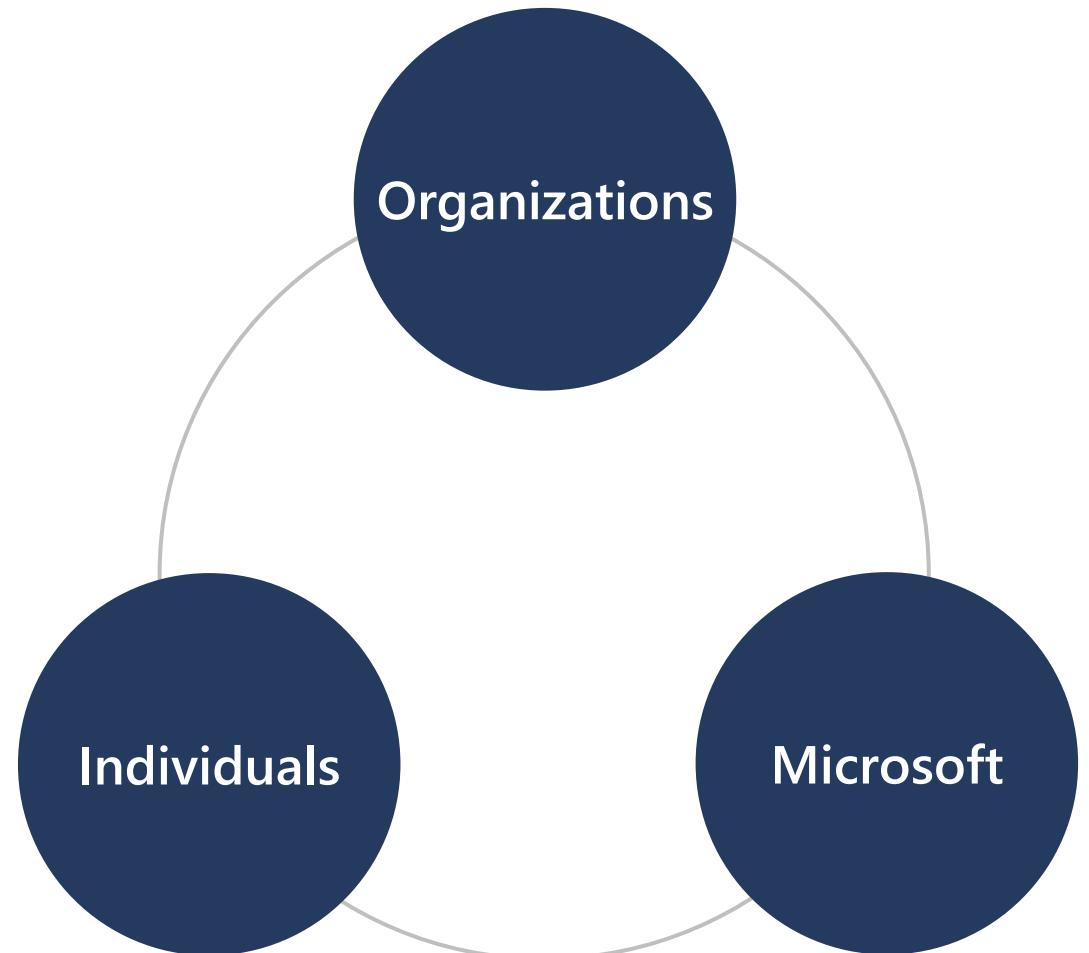


The Azure security center

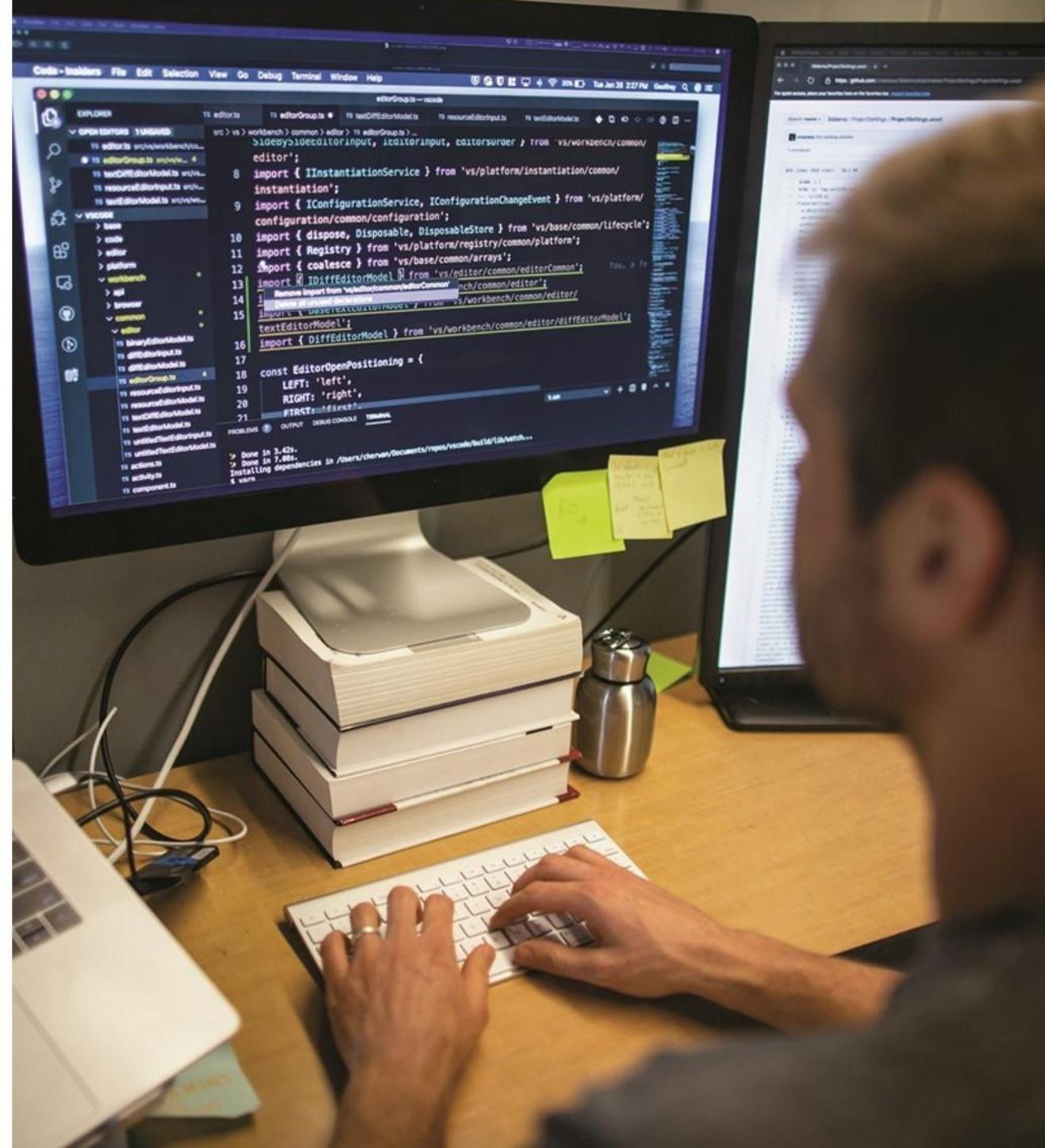
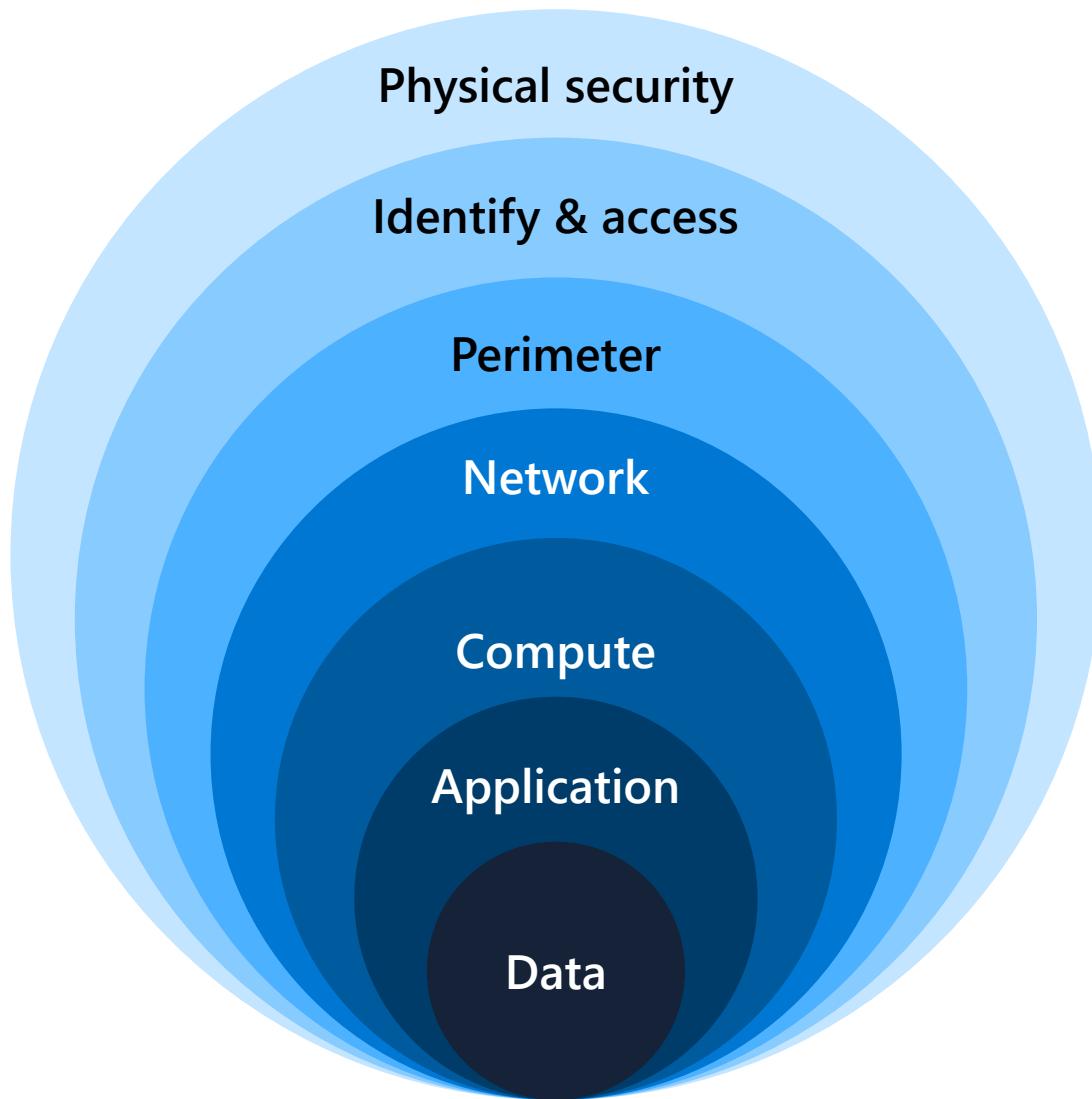


Azure Government

# Shared security responsibility



# A layered approach to security



# Azure security center

The screenshot shows the Microsoft Azure Security Center landing page. At the top, there's a dark header with the Microsoft Azure logo and a navigation bar with links: Overview, Solutions, Products (underlined), Documentation, Pricing, Training, Marketplace, and Partners. Below the header, the main title "Azure Security Center" is displayed in large white font. A sub-headline "Gain unmatched hybrid security management and threat protection" follows. A prominent blue button labeled "Turn on Security Center >" is centered. Below it, a message for non-subscribers "Not yet subscribed to Azure? Start free >" is shown. At the bottom of the main section, there are links for Pricing, Documentation, Updates, and Training. A call-to-action box at the bottom left encourages users to "Turn on protection you need". To its right, a text box explains that Microsoft uses various actions to help safeguard security posture against threats.

Microsoft Azure

Overview Solutions **Products** Documentation Pricing Training Marketplace Partners

## Azure Security Center

Gain unmatched hybrid security management and threat protection

Turn on Security Center >

Not yet subscribed to Azure? [Start free >](#)

Pricing > Documentation > Updates > Training >

Turn on protection you need

Microsoft uses a wide variety of physical, infrastructure, and additional actions you need to take to help safeguard your security posture and protect against threats.

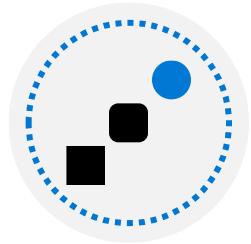
## Use for incident response:

You can use Security Center during the detection, assessment, and diagnosis of security at various stages

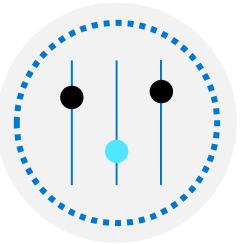
## Use to enhance security:

Reduce the chances of a significant security event by configuring a security policy, and then implementing the recommendations provided by Azure Security Center

# Azure Government



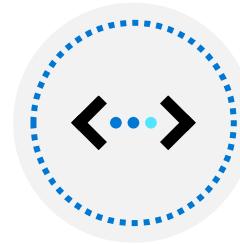
**Modernize  
Government  
services**



**Provide a  
platform of agility**



**Advanced  
Government  
mission**



**Physically separate  
from Azure**

## Lesson 02: Key security components



## Lesson objectives



Network security



Identity and access management



Encryption capabilities built into Azure



Azure threat protection

# 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
<p>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</p>	<p>To provide inbound protection at the perimeter, there are several choices:</p> <ul style="list-style-type: none"><li>• Azure Firewall</li><li>• Azure Application Gateway</li><li>• Azure Storage Firewall</li></ul>	<p>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</p>	<p>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</p>

# 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 provides 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 organization'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

# Azure threat protection

Azure Advanced Threat Protection | contoso-corp | Timeline PREVIEW

4:04 PM Today

**Honeytoken activity** Updated

The following activities were performed by [Bob Minion](#):

- Logged in to 2 computers via Contoso-DC.
- Authenticated from 2 computers using Kerberos when accessing 5 resources against Contoso-DC.
- Authenticated from iTARGOET-T4705 using NTLM against corporate resources via Contoso-DC.

Started at 3:08 PM Jan 22, 2018

3:23 PM Jan 22, 2018

**Remote execution attempt detected**

The following remote execution attempts were performed on Contoso-DC from ALICE-DESKTOP:

- Attempted remote execution of one or more WMI methods by AdminUser.

3:06 PM Jan 22, 2018

**Suspicious service creation**

AdminUser created 10 services in order to execute potentially malicious commands on Contoso-DC.

3:03 PM Jan 22, 2018

**Brute force attack using LDAP simple bind**

200 password guess attempts were made on 2 accounts from ALICE-DESKTOP. 2 account passwords were successfully guessed.

2:59 PM Jan 22, 2018

**Reconnaissance using account enumeration**

Suspicious account enumeration activity using Kerberos protocol, originating from ALICE-DESKTOP, was detected. The attacker performed a total of 101 guess attempts for account names. 2 guess attempts matched existing account names in Active Directory.

12:38 PM Jan 21, 2018

**Malicious replication of directory services**

Malicious replication requests were attempted by Alice Liddle from ALICE-DESKTOP against Contoso-DC.

11:59 AM Jan 21, 2018

**Reconnaissance using DNS**

Suspicious DNS activity was observed, originating from ALICE-DESKTOP (which is not a DNS server) against Contoso-DC.

This screenshot shows the Azure Advanced Threat Protection Timeline interface. It displays a list of threat events with their times, descriptions, and details. The events include Honeytoken activity, Remote execution attempt detected, Suspicious service creation, Brute force attack using LDAP simple bind, Reconnaissance using account enumeration, Malicious replication of directory services, and Reconnaissance using DNS. Each event has an 'OPEN' button and a more options menu. The interface is clean with a white background and blue accents for links.

## Lesson 03: Securing storage accounts and Data Lake Storage



## Lesson objectives



Storage account security features



**Explore the authentication options available to access data:** Storage account key | Shared access signature



Control network access to the data



Managing encryption



Azure Data Lake Storage Gen II security features

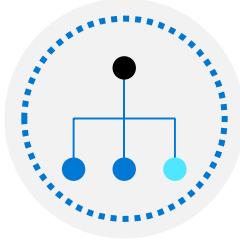
# Storage account security features



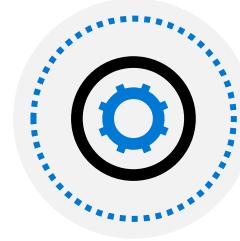
Encryption  
at rest



Encryption  
in transit



Role based  
access control



Auditing  
access

# Storage account keys

Home > Resource groups > cto\_rg > ctoazureblob - Access keys

## ctoazureblob - Access keys

Storage account

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Storage Explorer (preview)

Settings

Access keys

Geo-replication

CORS

Configuration

Encryption

Shared access signature

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure K Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will interrupt access to disks from your virtual machines. [Learn more](#)

Storage account name  
ctoazureblob

key1

Key  
eU7 [REDACTED] Cg==

Connection string  
Def [REDACTED] 9YrQ...

key2

Key  
NW0 [REDACTED] A/pUgB5w==

Connection string  
Def [REDACTED] Ns6...

# Shared access signatures

Home > Resource groups > cto\_rg > ctoazureblob - Shared access signature

## ctoazureblob - Shared access signature

A storage account

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Storage Explorer (preview)

**Settings**

Access keys

Geo-replication

CORS

Configuration

Encryption

**Shared access signature**

Firewalls and virtual networks

Advanced Threat Protection

Static website

Properties

Locks

Export template

Blob service

A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you grant them access to a resource for a specified period of time.

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

[Learn more](#)

Allowed services ⓘ

Blob  File  Queue  Table

Allowed resource types ⓘ

Service  Container  Object

Allowed permissions ⓘ

Read  Write  Delete  List  Add  Create  Update  Process

Start and expiry date/time ⓘ

Start  
2019-03-29  11:59:33

End  
2019-03-29  19:59:33

(UTC+00:00) --- Current Time Zone ---

Allowed IP addresses ⓘ  
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols ⓘ

HTTPS only  HTTPS and HTTP

Signing key ⓘ

key1

**Generate SAS and connection string**

# Control network access to data

## Firewalls and virtual networks

Save Discard Refresh

i Firewall settings allowing access to storage services will remain in effect for up to a minute after saving updated settings restricting access.

Allow access from  All networks  Selected networks

Configure network security for your storage accounts. [Learn more.](#)

Virtual networks  
Secure your storage account with virtual networks. [+ Add existing virtual network](#) [+ Add new virtual network](#)

VIRTUAL NETWORK	SUBNET	ADDRESS RANGE	ENDPOINT STATUS	RESOURCE GROUP	SUBSCRIPTION
No network selected.					

Firewall  
Add IP ranges to allow access from the internet or your on-premises networks. [Learn more.](#)

Add your client IP address ('86.184.235.180') i

**ADDRESS RANGE**

Exceptions

Allow trusted Microsoft services to access this storage account i

Allow read access to storage logging from any network

Allow read access to storage metrics from any network

# Managing encryption

Databases stores information that is sensitive, such as physical addresses, email addresses, and phone numbers. The following can be used to protect this data

## Transport Layer Security (TLS)

Azure SQL Database and Data Warehouse enforces Transport Layer Security (TLS) encryption at all times for all connections, which ensures all data is encrypted "in transit" between the database and the client

## Transparent data encryption

Both Azure Data Warehouse and SQL Database protects your data at rest using transparent data encryption (TDE). TDE performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application

## Application encryption

Data in transit is a method to prevent man-in-the-middle attacks. To encrypt data in transit, specify **Encrypt=true** in the connection string in your client applications

# Azure Data Lake Storage Gen2 security features

Role based  
access  
control

POSIX  
compliant  
ACL

AAD Oauth  
2.0 token

Azure  
services  
integration

Encryption

## Lesson 04: Securing data stores



## Lesson objectives



Control network access to your data stores using firewall rules



Control user access to your data stores using authentication and authorization



Dynamic data masking



Audit and monitor your Azure SQL Database for access violations

# Control network access to your data stores using firewall rules

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

These rules enable clients to access your **entire Azure SQL server**, that is, all the databases within the same logical server

## Database level firewall rules

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

# Control user access to your data stores using authentication and authorization

## Authentication

SQL Database and Azure Synapse Analytics supports two types of authentication: SQL authentication and Azure Active Directory authentication

## Authorization

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

# Dynamic data masking

## Masking rules

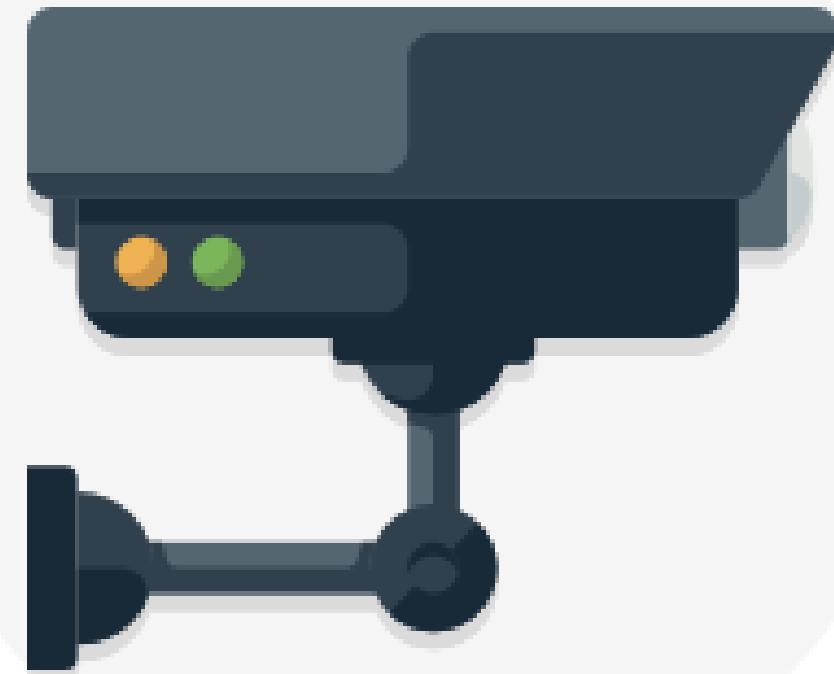
MASK NAME	MASK FUNCTION
You haven't created any masking rules.	
SQL users excluded from masking (administrators are always excluded) <small>i</small>	

## Recommended fields to mask

SCHEMA	TABLE	COLUMN	
SalesLT	Address	AddressID	<button>Add mask</button>
SalesLT	Address	AddressLine1	<button>Add mask</button>
SalesLT	Address	AddressLine2	<button>Add mask</button>
SalesLT	Customer	FirstName	<button>Add mask</button>
SalesLT	Customer	LastName	<button>Add mask</button>

[Load more](#)

# Auditing and monitoring



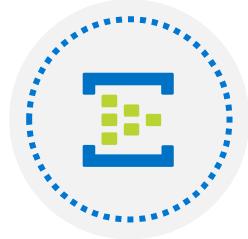
## Lesson 05: Securing streaming data



## Lesson objectives



Understand Stream Analytics security



Understand Event Hub security

# Stream Analytics security

## Data in transit

Azure Stream Analytics encrypts all incoming and outgoing communications and supports Transport Layer Security v 1.2

## Data at rest

Stream Analytics doesn't store the incoming data since all processing is done in-memory. Therefore, consider setting security for services such as Event Hubs or Internet of Things Hubs, or for data stores such as Cosmos DB

# Event Hub security

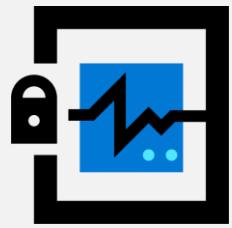
## Authentication

Authentication makes use of Shared Access Signatures and Event Publishers to ensure that only applications or devices with valid credentials are only allowed to send data to an Event Hub. Each client is assigned a token

## Token management

Once the tokens have been created, each client is provisioned with its own unique token. If a token is stolen by an attacker, the attacker can impersonate the client whose token has been stolen. Adding a client to a blocked recipients list renders that client unusable

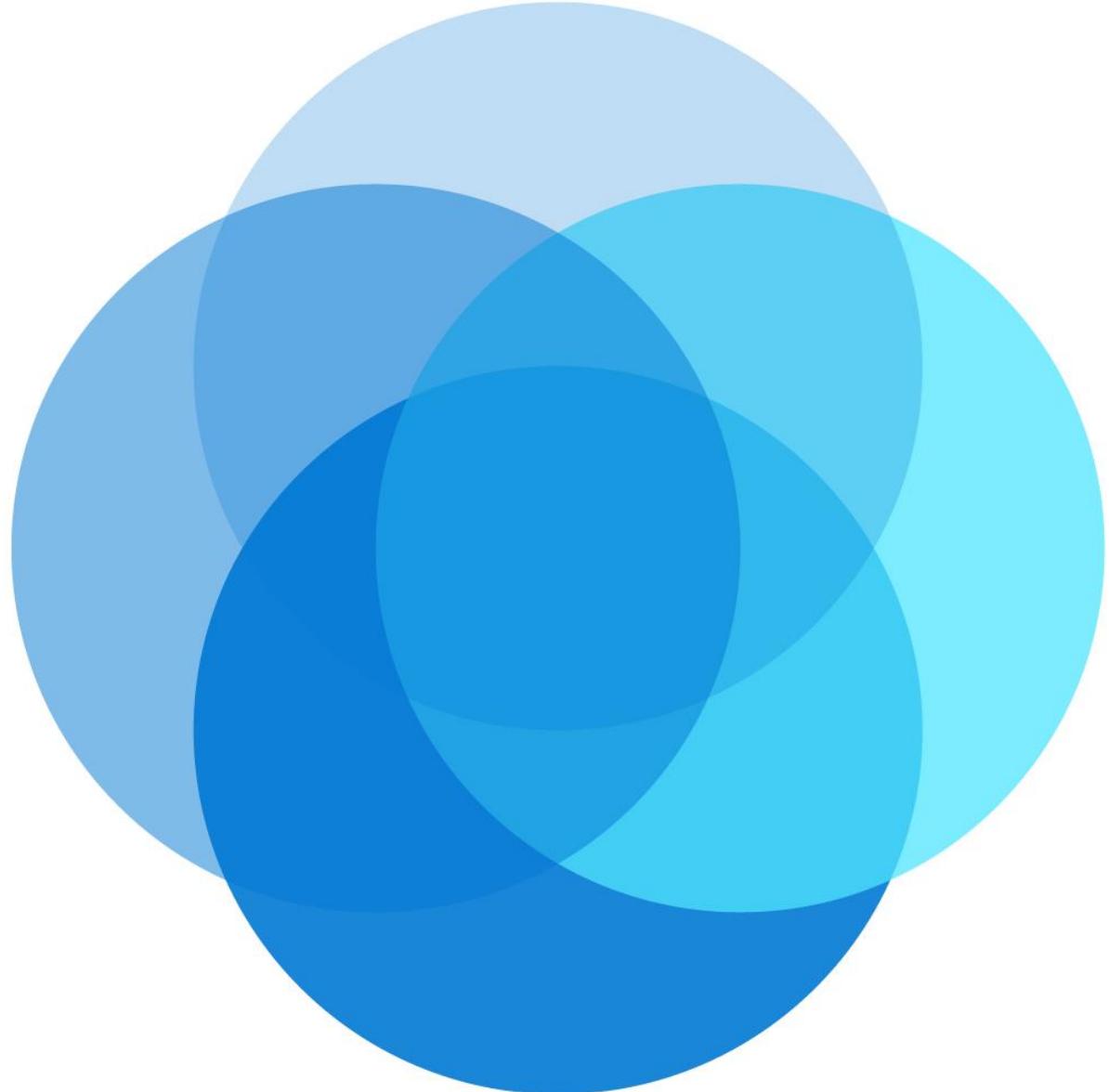
# Lab: Securing Azure Data Platforms





# Module 09:

## Monitoring and troubleshooting data storage and processing

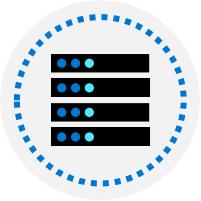


# Agenda



Lesson 01: General Azure monitoring capabilities

---



Lesson 02: Troubleshoot common data storage issues

---



Lesson 03: Troubleshoot common data processing issues

---



Lesson 04: Manage disaster recovery

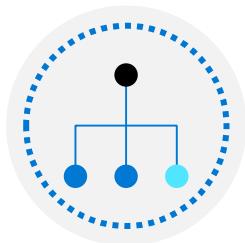
# Lesson 01: General Azure monitoring capabilities



## Lesson objectives



Azure Monitor



Monitoring the network



Diagnose and solve problems

# Azure Monitor

Azure Monitor provides a holistic monitoring approach by collecting, analyzing, and acting on telemetry from both cloud and on-premises environments

## Metric data

Provides quantifiable information about a system over time that enables you to observe the behavior of a system

## Log data

Logs can be queried and even analyzed using Azure Monitor logs. In addition, this information is typically presented in the overview page of an Azure Resource in the Azure portal

## Alerts

Alerts notify you of critical conditions and potentially take corrective automated actions based on triggers from metrics or logs

# Monitoring the network

Azure Monitor logs within Azure monitor has the capability to monitor and measure network activity

## Network performance monitor

Network Performance Monitor measures the performance and reachability of the networks that you have configured

## Application gateway analytics

Application Gateway Analytics contains rich, out-of-the box views you can get insights into key scenarios, including:

- Monitor client and server errors
- Check requests per hour

# Diagnose and solve issues

Home > ctocdb - Diagnose and solve problems

## ctocdb - Diagnose and solve problems

Azure Cosmos DB account

x

[Overview](#)[Activity log](#)[Access control \(IAM\)](#)[Tags](#)[Diagnose and solve problems](#)[Quick start](#)[Notifications](#)[Data Explorer](#)

---

### SETTINGS

[Replicate data globally](#)[Default consistency](#)[Firewall and virtual networks](#)[CORS](#)

---

### RESOURCE HEALTH

Available

There aren't any known problems affecting this Cosmos DB database account [More details](#)

### RECENT ACTIVITY

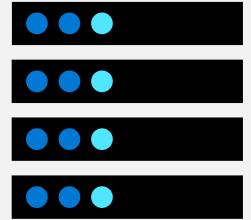
Activity for the past 24 hours

[Quick Insights](#) | [See all activity](#)

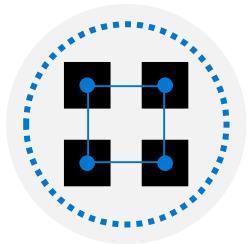
### SOLUTIONS TO COMMON PROBLEMS

- ✓ My database is slow
- ✓ My request unit (RU) charging is unclear
- ✓ I need more storage/throughput
- ✓ My queries are slow
- ✓ MongoDB API Support
- ✓ Import MongoDB data into CosmosDB

## Lesson 02: Troubleshoot common data storage issues



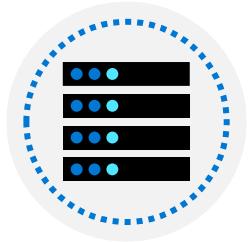
## Lesson objectives



Connectivity issues



Performance issues



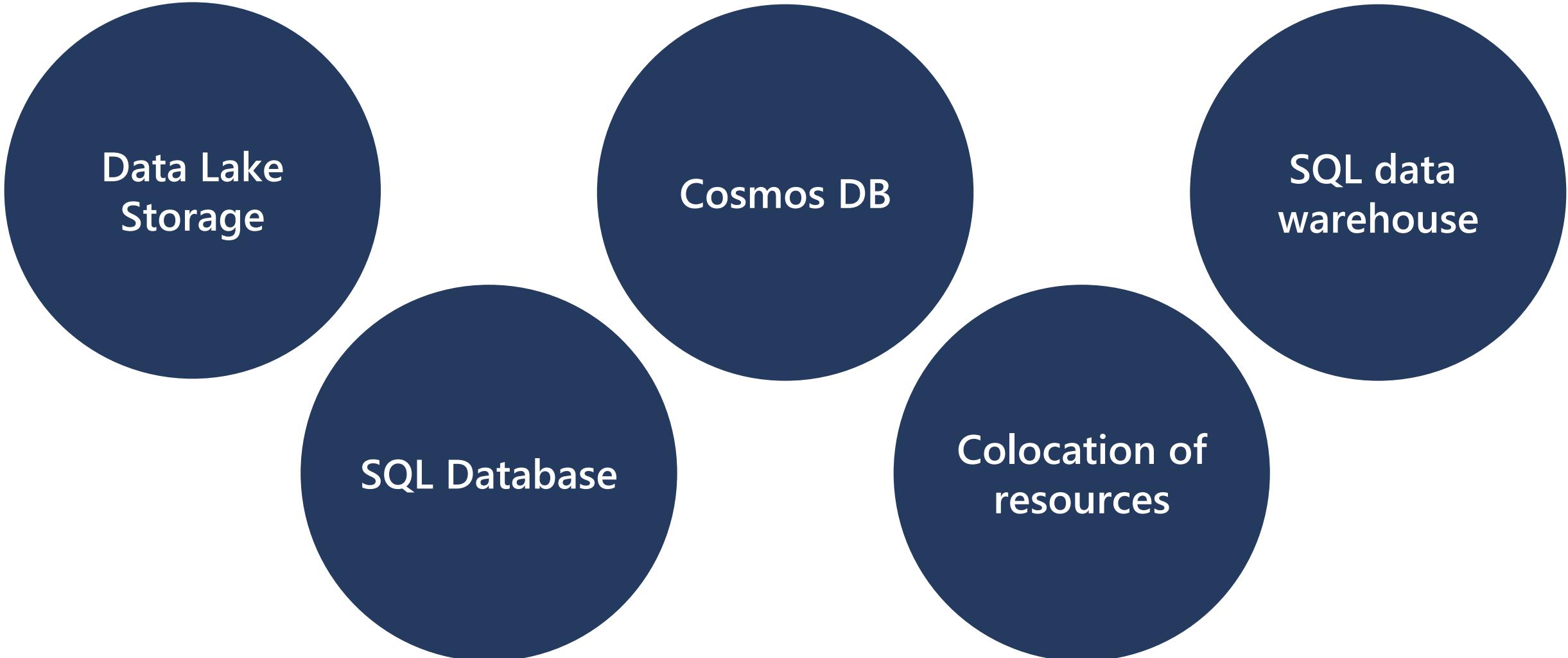
Storage issues

# Connectivity issues

There are a range of issues that can impact connectivity issues, including:

Unable to connect to the data platform	Authentication failures	Cosmos DB Mongo DB API errors	SQL database failover
<p>The first area that you should check is the firewall configuration</p> <p>Test the connection by accessing it from a location external to your network</p> <p>Check maintenance schedules</p>	<p>The first check is to ensure that the username and password is correct</p> <p>Check the storage account keys and ensure that they match in the connection string</p>	<p>Mongo client drivers establishes more than one connection</p> <p>On the server side, connections which are idle for more than 30 minutes are automatically closed down</p> <p>Check for timeouts</p>	<p>Should you receive an “unable to connect” message (error code 40613) in the Azure SQL Database, this scenario commonly occurs when a database has been moved because of deployment, failover, or load balancing</p>

# Performance issues



# Storage issues

## Consistency:

Consider the consistency levels of the following data stores that can impact data consistency:

- Cosmos DB
- SQL Data Warehouse
- SQL Database

## Corruption:

Data corruption can occur on any of the data platforms for a variety of reasons. You should have an appropriate disaster recovery strategy

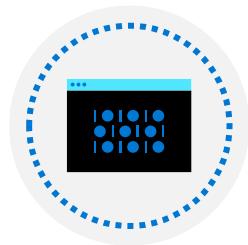
## Lesson 03: Troubleshoot common data processing issues



## Lesson objectives



Troubleshoot streaming data



Troubleshoot batch data loads



Troubleshoot Azure Data Factory

# Troubleshoot streaming data

When using Stream Analytics, a Job encapsulates the Stream Analytic work and is made up of three components:

## Job input

The job input contains a **Test Connection** button to validate that there is connectivity with the input. However, most errors associated with a job input is due to the malformed input data that is being ingested

## Job query

A common issue associated with Stream Analytics query is the fact that the output produced is not expected. In this scenario it is best to check the query itself to ensure that there is no mistakes on the code there

## Job output

As with the job input, there is a \*\*Test Connection\*\* button to validate that there is connectivity with the output, should there be no data appearing. You can also use the \*\*Monitor\*\* tab in Stream Analytics to troubleshoot issues

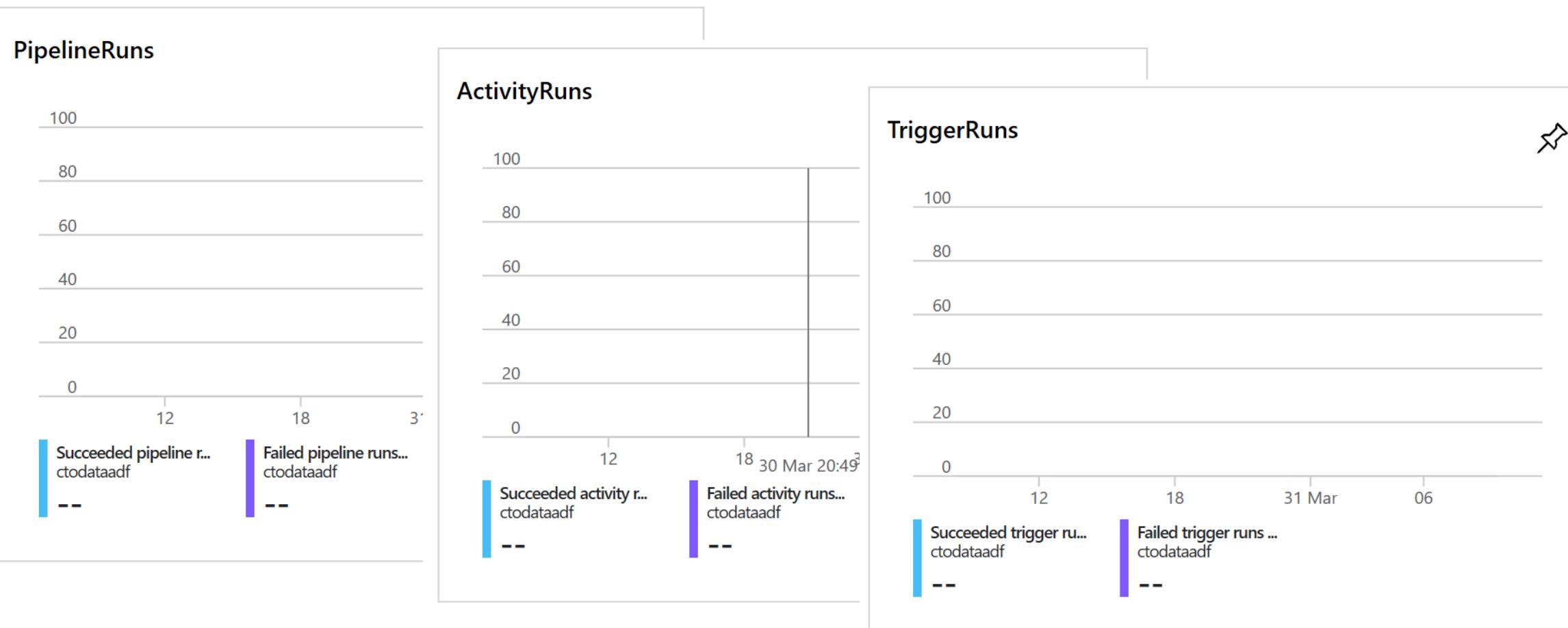
# Troubleshoot batch data loads

When trying to resolve data load issues, it is first pragmatic to make the holistic checks on Azure, as well as the network checks and diagnose and solve issue check. After that, then check:

Azure Blob and Data Lake Store	SQL Data Warehouse	Cosmos DB	SQL Database
Notwithstanding network errors; occasionally, you can get timeout or throttling errors that can be a symptom of the availability of the storage accounts	Make sure you are always leveraging PolyBase  Ensure CTAS statements are used to load data  Break data down into multiple text files  Consider DWU usage	Check that you have provisioned enough RU's  Review partitions and partitioning keys  Check for client connection string settings	Check that you have provisioned enough DTU's  Review whether the database would benefit from elastic pools  A wide range of tools can be used to troubleshoot SQL Database

# Troubleshoot Azure Data Factory

## Monitoring



## Lesson 04: Managing disaster recovery



## Lesson objectives



Data redundancy



Disaster recovery

# Data redundancy

Data redundancy is the process of storing data in multiple locations to ensure that it is highly available

Azure Blob and Data Lake Store	SQL Data Warehouse	Cosmos DB	SQL Database
Locally redundant storage (LRS) Zone-redundant storage (ZRS) Geo-redundant storage (GRS) Read-access geo-redundant storage (RA-GRS)	SQL Data Warehouse performs a <a href="#">geo-backup</a> once per day to a paired data center. The RPO for a geo-restore is 24 hours	Azure Cosmos DB is a globally distributed database service. You can configure your databases to be globally distributed and available in any of the Azure regions	Check that you have provisioned enough DTU's  Review whether the database would benefit from elastic pools  A wide range of tools can be used to troubleshoot SQL Database

# Disaster Recovery

There should be processes that are involved in backing up or providing failover for databases in an Azure data platform technology. Depending on circumstances, there are numerous approaches that can be adopted

Azure Blob and Data Lake Store	SQL Data Warehouse	Cosmos DB	SQL Database
Supports account failover for geo-redundant storage accounts  You can initiate the failover process for your storage account if the primary endpoint becomes unavailable	SQL Data Warehouse performs a <b>geo-backup</b> once per day to a paired data center  Data warehouse snapshot feature that enables you to create a restore point to create a copy of the warehouse to a previous state	Takes a backup of your database every <b>4 hours</b> and at any point of time  Only the latest 2 backups are stored	Creates database backups that are kept between 7 and 35 days  Uses Azure read-access geo-redundant storage (RA-GRS) to ensure that they preserved even if data center is unavailable

## Lab: Monitoring and troubleshooting data storage and processing

