Microsoft

DP-200T01: Implementing an Azure Data Solution

Tissana Tanaklang

Software and Solution Development Trainer
Iverson Training Center Co., Ltd.
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 Solutions Associate (MCSA) - Web Application Development
- Microsoft Certified Azure Data Fundamentals
- Microsoft Certified Azure Fundamentals
- Microsoft Certified Trainer (MCT)

Azure Learning Path

Level	Category	Code	Course	Role
Beginner (Fundamentals)	-	AZ-900	Microsoft Azure Fundamentals	IT Professional and Non-IT Professional (All)
	Data	DP-900	Microsoft Azure Data Fundamentals	Data Engineer, Database Administrator
	AI	AI-900	Microsoft Azure AI Fundamentals	AI Engineer, Data Scientist, Developer, Solutions Architect
Intermediate (Associate)	DevOps	AZ-104	Microsoft Azure Administrator	Administrator, DevOps Engineer
		AZ-204	Developing solutions for Microsoft Azure	Developer, DevOps Engineer
	Security	AZ-500	Microsoft Azure Security Technologies	Security Engineer
	Data	DP-300	Administering Relational Databases on Microsoft Azure	Database Administrator
		DP-200	Implementing an Azure Data Solution	Data Engineer
		DP-201	Designing an Azure Data Solution	
		DP-100	Designing and Implementing a Data Science Solution on Azure	Data Scientist
	AI	AI-100	Designing and Implementing an Azure AI Solution	AI Engineer
Advance (Expert)	DevOps	AZ-400	Designing and Implementing Microsoft DevOps solutions	DevOps Engineer
	Solutions Architect	AZ-303	Microsoft Azure Architect Technologies	Solutions Architect
		AZ-304	Microsoft Azure Architect Design	
Specialty	Data	DA-100	Analyzing Data with Power BI	Data Analyst
	-	AZ-220	Microsoft Azure IoT Developer	Developer

Microsoft

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
  - L01 - Explain the evolving world of data
  - L02 - Survey the services in the Azure Data Platform
  - L03 - Identify the tasks that are performed by a Data Engineer
  - L04 - Describe the use cases for the cloud in a case study

- Module 2
- Working with Data Storage
  - L01 - Choose a data storage approach in Azure
  - L02 - Create an Azure Storage Account
  - L03 - Explain Azure Data Lake Storage
  - L04 - Upload data into Azure Data Lake

Course Agenda (*continued #1*)

- Module 3
- Enabling Team Based Data Science with Azure Databricks
  - L01 - Explain Azure Databricks
  - L02 - Work with Azure Databricks
  - L03 - Read data with Azure Databricks
  - L04 - Perform transformations with Azure Databricks

- Module 4
- Building Globally Distributed Databases with Cosmos DB
  - L01 - Create an Azure Cosmos DB database built to scale
  - L02 - Insert and query data in your Azure Cosmos DB database
  - L03 - Build a .NET Core app for Azure Cosmos DB in Visual Studio Code
  - L04 - Distribute your data globally with Azure Cosmos DB

Course Agenda (*continued #2*)

- Module 5
- Working with Relational Data Stores in the Cloud
  - L01 - Explain SQL Database
  - L02 - Explain SQL Data Warehouse
  - L03 - Provision and load data in Azure SQL Data Warehouse
  - L04 - Import data into Azure SQL Data Warehouse using PolyBase

- Module 6
- Performing Real-Time Analytics with Stream Analytics
  - L01 - Explain data streams and event processing
  - L02 - Data Ingestion with Event Hubs
  - L03 - Processing Data with Stream Analytics Jobs

Course Agenda (*continued #3*)

- Module 7
- Orchestrating Data Movement with Azure Data Factory
  - L01 - Explain how Azure Data Factory works
  - L02 - Create Linked Services and Datasets
  - L03 - Create Pipelines and Activities
  - L04 - Azure Data Factory pipeline execution and triggers

- Module 8
- Securing Azure Data Platforms
  - L01 - Introduction to Security
  - L02 - Key Security Components
  - L03 - Securing Storage Accounts and Data Lake Storage
  - L04 - Security Data Stores

Course Agenda (*continued #4*)

- Module 9
- Monitoring and Troubleshooting Data Storage and Processing
  - L01 - Explain the monitoring capabilities that are available
  - L02 - Troubleshoot common data storage issues
  - L03 - Troubleshoot common data processing issues
  - L04 - 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

Microsoft

Module 01: Azure for the Data Engineer

Agenda

- L01 – Explain the evolving world of data
- L02 - Survey the services in the Azure Data Platform
- L03 - Identify the tasks that are performed by a Data Engineer
- L04 - 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

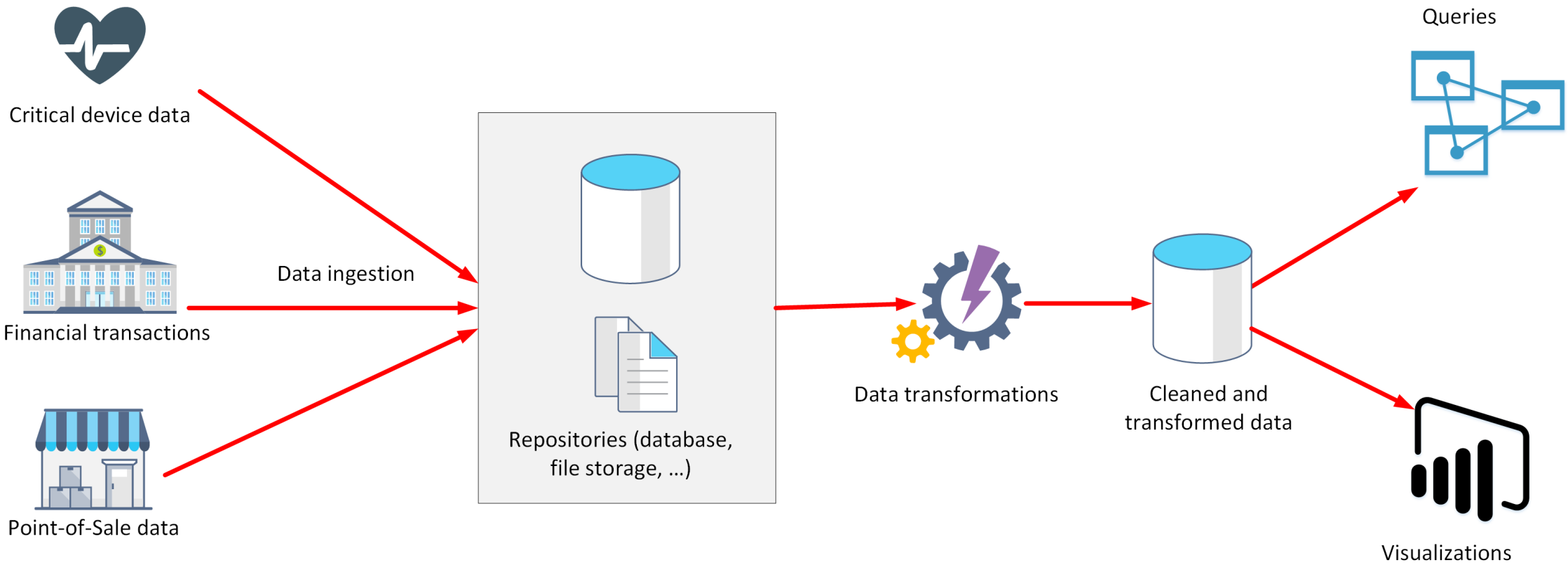
A data engineers role is responsible for more data types and technologies

Technologies

Microsoft Azure provides a wide set of tools and technologies

On-premises versus cloud technologies

Data engineering job responsibilities


Critical device data
Financial transactions
Point-of-Sale data
Data ingestion
Repositories (database, file storage, …)
Data transformations
Cleaned and transformed data
Queries
Visualizations

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

	Schema	Data relationships	Examples
Structured data	Adheres to a schema, with the same data fields or properties.	Storable in relational database tables, with rows and columns.	Sensor data and financial data.
Semi-structured data	Has an ad hoc schema with less organized fields and properties.	Non-relational or NoSQL data, not storable in tables, rows and column.	Books, blogs, JSON, HTML documents.
Unstructured data	Has no designated schema or data structure.	Non-relational or blob data, with no restrictions on the kinds of data blobs contain.	PDFs, JPGs, videos.

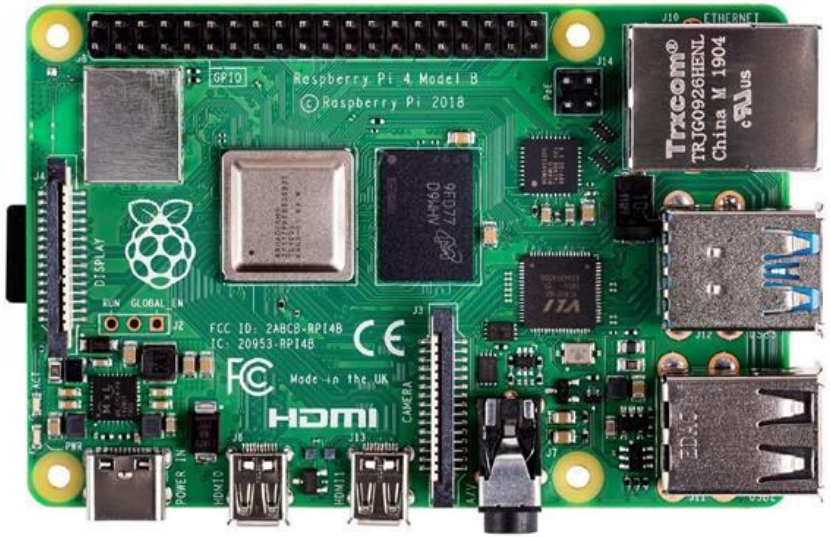
Data Categories

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	Semi- Structured	Unstructured
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 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 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

- IoT and telematics.
- Retail and marketing.
- Web and mobile applications.
- Gaming



Non-Relational Database use case

{"latitude":37.8267,"longitude":-122.4233,"timezone":"America/Los_Angeles","currently":{"time":1598191217,"summary":"Partly Cloudy","icon":"partly-cloudy-
day","nearestStormDistance":5,"nearestStormBearing":58,"precipIntensity":0,"precipProbability":0,"temperature":58.63,"apparentTemperature":58.63,"dew
Point":52.42,"humidity":0.8,"pressure":1011.8,"windSpeed":5.08,"windGust":7.73,"windBearing":210,"cloudCover":0.54,"uvIndex":0,"visibility":9.933,"oz
one":291.2},"minutely":{"summary":"Partly cloudy for the hour.","icon":"partly-cloudy-day","data":
[{"time":1598191200,"precipIntensity":0,"precipProbability":0},{"time":1598191260,"precipIntensity":0,"precipProbability":0},
{"time":1598191320,"precipIntensity":0,"precipProbability":0},{"time":1598191380,"precipIntensity":0,"precipProbability":0},
{"time":1598191440,"precipIntensity":0,"precipProbability":0},{"time":1598191500,"precipIntensity":0,"precipProbability":0},
{"time":1598191560,"precipIntensity":0,"precipProbability":0},{"time":1598191620,"precipIntensity":0,"precipProbability":0},
{"time":1598191680,"precipIntensity":0,"precipProbability":0},{"time":1598191740,"precipIntensity":0,"precipProbability":0},
{"time":1598191800,"precipIntensity":0,"precipProbability":0},{"time":1598191860,"precipIntensity":0,"precipProbability":0},
{"time":1598191920,"precipIntensity":0,"precipProbability":0},{"time":1598191980,"precipIntensity":0,"precipProbability":0},
{"time":1598192040,"precipIntensity":0,"precipProbability":0},{"time":1598192100,"precipIntensity":0,"precipProbability":0},
{"time":1598192160,"precipIntensity":0,"precipProbability":0},{"time":1598192220,"precipIntensity":0,"precipProbability":0},
{"time":1598192280,"precipIntensity":0,"precipProbability":0},{"time":1598192340,"precipIntensity":0,"precipProbability":0},
{"time":1598192400,"precipIntensity":0.0026,"precipIntensityError":0.0004,"precipProbability":0.01,"precipType":"rain"},
{"time":1598192460,"precipIntensity":0.0026,"precipIntensityError":0.0005,"precipProbability":0.01,"precipType":"rain"},
{"time":1598192520,"precipIntensity":0,"precipProbability":0},
{"time":1598192580,"precipIntensity":0.0026,"precipIntensityError":0.0005,"precipProbability":0.01,"precipType":"rain"},
{"time":1598192640,"precipIntensity":0.0026,"precipIntensityError":0.0005,"precipProbability":0.01,"precipType":"rain"},
{"time":1598192700,"precipIntensity":0.0027,"precipIntensityError":0.0005,"precipProbability":0.01,"precipType":"rain"},
{"time":1598192760,"precipIntensity":0.0027,"precipIntensityError":0.0005,"precipProbability":0.02,"precipType":"rain"},
{"time":1598192820,"precipIntensity":0.0026,"precipIntensityError":0.0005,"precipProbability":0.02,"precipType":"rain"},
{"time":1598192880,"precipIntensity":0,"precipProbability":0},

Non-Relational Database use case

https://api.darksky.net/

Open API :] สำหรับนักพัฒนา

แสดงค่าประจำวัน :
//covid19.th-stat.com/api/open/today

ข้อมูลสรุปตามช่วงเวลา (เริ่มตั้งแต่วันที่ 01/01/20) :
//covid19.th-stat.com/api/open/timeline

ข้อมูลแต่ละเคส :
//covid19.th-stat.com/api/open/cases

ข้อมูลสรุปจากเคส :
//covid19.th-stat.com/api/open/cases/sum

แจ้งเตือนพื้นที่ตามคำประกาศ :
//covid19.th-stat.com/api/open/area

กรมควบคุมโรค
DEPARTMENT OF DISEASE CONTROL

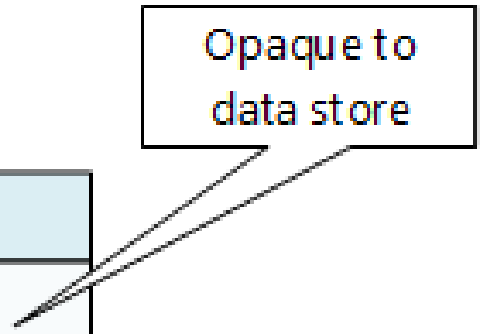
Non-Relational Database use case

- You might see the term *NoSQL* when reading about non-relational databases.
- NoSQL is a rather loose term that simply means non-relational.
- NoSQL (non-relational) databases generally fall into four categories:

  - key-value stores
  - document databases
  - column family databases
  - graph databases.

What is NoSQL?

Key	Value
AAAAA	1101001111010100110101111...
AABAB	1001100001011001101011110...
DFA766	0000000000101010110101010...
FABCC4	1110110110101010100101101...



A key-value store is the simplest (and often quickest) type of NoSQL database for inserting and querying data.

Key-Value Stores

Key	Document

1001	{
	"CustomerID": 99,
	"OrderItems": [
	{ "ProductID": 2010,
	"Quantity": 2,
	"Cost": 520
	},
	{ "ProductID": 4365,
	"Quantity": 1,
	"Cost": 18
	}],
	"OrderDate": "04/01/2017"
	}

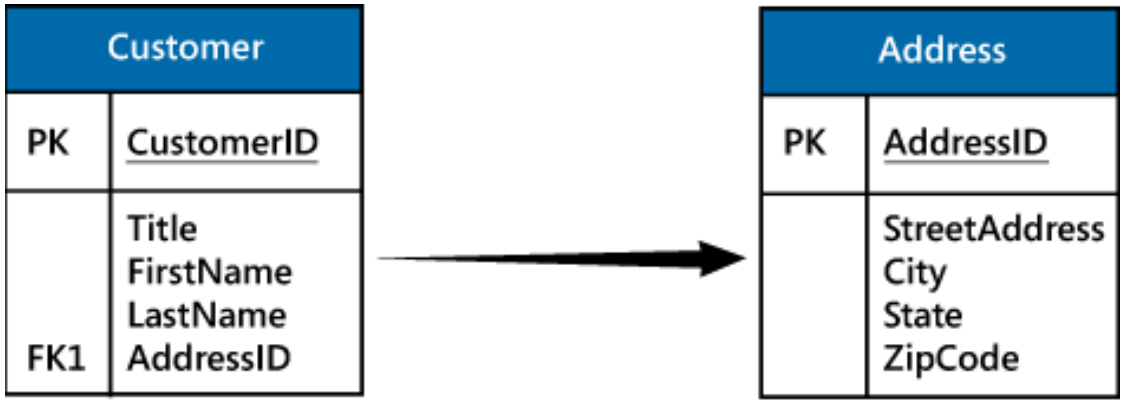
1002	{
	"CustomerID": 220,
	"OrderItems": [
	{ "ProductID": 1285,
	"Quantity": 1,
	"Cost": 120
	}],
	"OrderDate": "05/08/2017"
	}

A document database represents the opposite end of the NoSQL spectrum from a key-value store. In a document database, each document has a unique ID, but the fields in the documents are transparent to the database management system. Document databases typically store data in JSON format,

Document Databases

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)

Document Databases



RDBMS is Row-based oriented

Customer Table

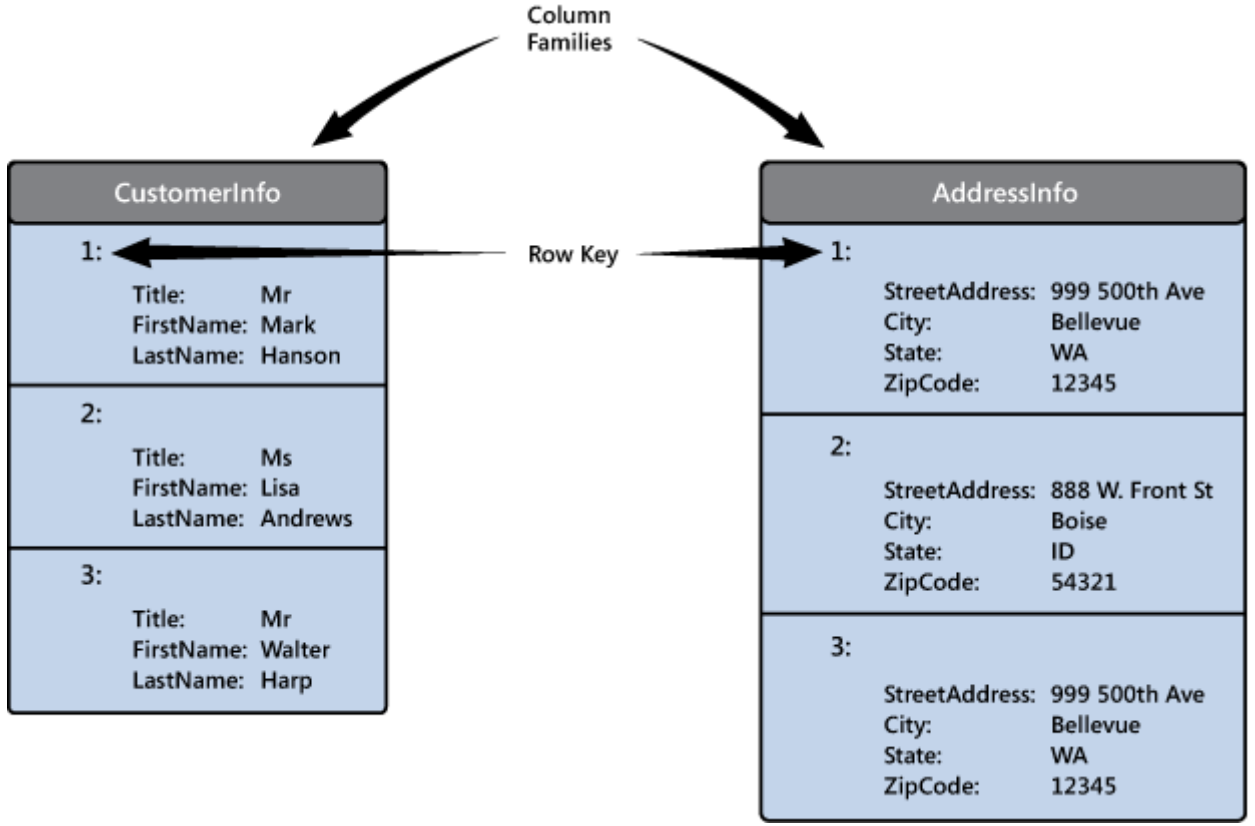
CustomerID	Title	FirstName	LastName	AddressID
1	Mr	Mark	Hanson	500
2	Ms	Lisa	Andrews	501
3	Mr	Walter	Harp	500

Address Table

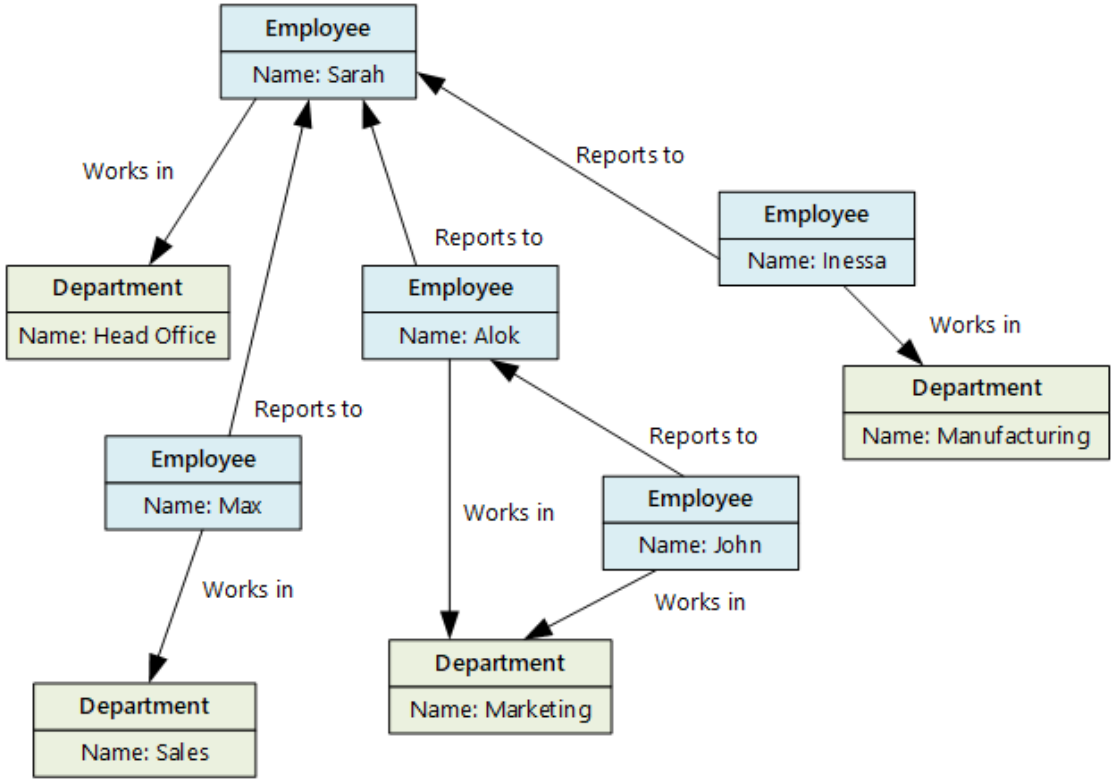
AddressID	StreetAddress	City	State	ZipCode
500	999 500th Ave	Bellevue	WA	12345
501	888 W. Front St	Boise	ID	54321

Column Family Databases

Row Key	Column Families	
CustomerID	CustomerInfo	AddressInfo
1	CustomerInfo:Title Mr CustomerInfo:FirstName Mark CustomerInfo:LastName Hanson	AddressInfo:StreetAddress 999 500th Ave AddressInfo:City Bellevue AddressInfo:State WA AddressInfo:ZipCode 12345
2	CustomerInfo:Title Ms CustomerInfo:FirstName Lisa CustomerInfo:LastName Andrews	AddressInfo:StreetAddress 888 W. Front St AddressInfo:City Boise AddressInfo:State ID AddressInfo:ZipCode 54321
3	CustomerInfo:Title Mr CustomerInfo:FirstName Walter CustomerInfo:LastName Harp	AddressInfo:StreetAddress 999 500th Ave AddressInfo:City Bellevue AddressInfo:State WA AddressInfo:ZipCode 12345



Column Family Databases



Graph databases enable you to store entities, but the main focus is on the relationships that these entities have with each other.

A graph database stores two types of information: nodes that you can think of as instances of entities, and edges, which specify the relationships between nodes.

Graph Databases

What to use for Data

Storage Account

Module 02

- When you need a **low cost, high throughput** data store.
- When you need to store **No-SQL** data.
- When you **do not need to query** the data directly. **No ad hoc query** support.
- Suits the storage of archive or **relatively static data.**
- Suits acting as a **HDInsight Hadoop** data store.

What to use for Data

Data Lake Store

Module 02

- When you need a **low cost, high throughput** data store.
- **Unlimited storage** for **No-SQL** data
- When you **do not need to query** the data directly. **No ad hoc query** support.
- Suits the storage of archive or **relatively static data.**
- Suits acting as a **Databricks** , **HDInsight** and **IoT** data store.

What to use for Data

Azure Databricks

Module 03

- **Eases the deployment** of a Spark based cluster.
- Enables the **fastest processing** of Machine Learning solutions.
- **Enables collaboration** between data engineers and data scientists.
- Provides **tight enterprise security integration** with Azure Active Directory
- **Integration with other Azure Services** and **Power BI.**

What to use for Data

Azure CosmosDB

Module 04

- Provides **global distribution** for both structured and unstructured data stores.
- **Millisecond query response** time.
- **99.999% availability** of data.
- **Worldwide elastic scale** of both the storage and throughput
- **Multiple consistency levels** to control data integrity with concurrency

What to use for Data

Azure SQL Database

Module 05

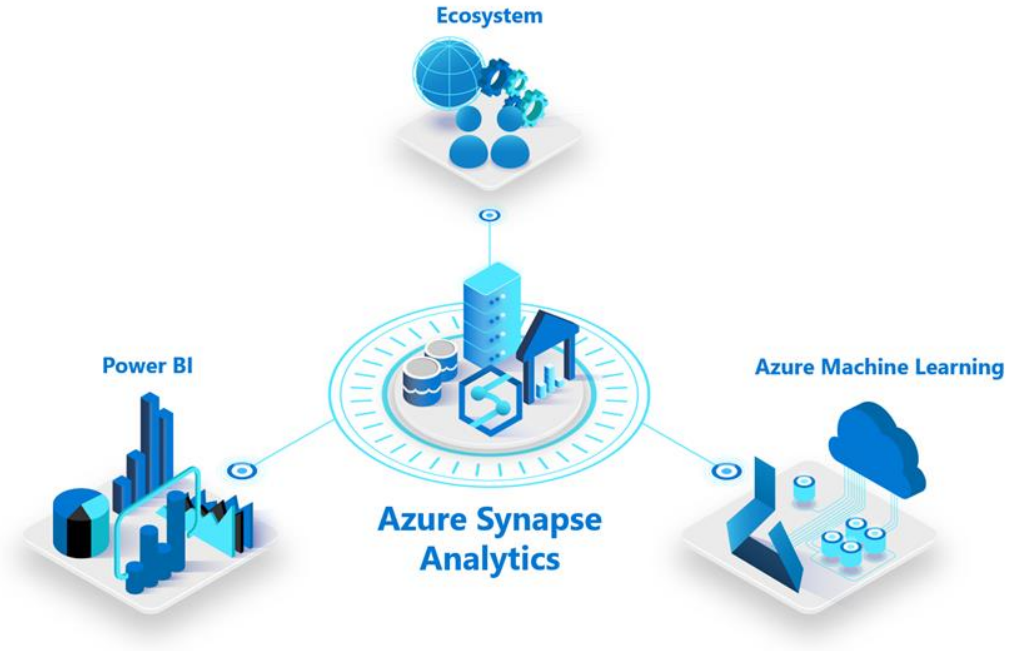
- When you require a **relational** data store.
- When you need to manage **transactional workloads**
- When you need to manage a **high volume on inserts and reads**
- When you need a service that **requires high concurrency**
- When you require a solution that can scale **elastically**

What to use for Data

Azure Synapse Analytics

Module 05

- When you require an integrated **relational** and **big data** store.
- When you need to manage **data warehouse** and **analytical workloads**
- When you need **low cost storage**.
- When you require the ability to **pause and restart the compute**.
- When you require a solution that can scale **elastically**



What to use for Data

Azure Stream Analytics

Module 06

- When you require a **fully managed event processing** engine.
- When you require **temporal analysis of streaming** data.
- Support for analyzing **IoT streaming** data.
- Support for analyzing application data through **Event Hubs.**
- Ease of use with a **Stream Analytics Query Language**.

What to use for Data

Azure Data Factory

Module 07

- When you want to **orchestrate the batch movement** of data.
- When you want to connect to **wide range of data platforms**.
- When you want to **transform or enrich** the data in movement.
- When you want to **integrate with SSIS packages**.
- Enables **verbose logging** of data processing activities.

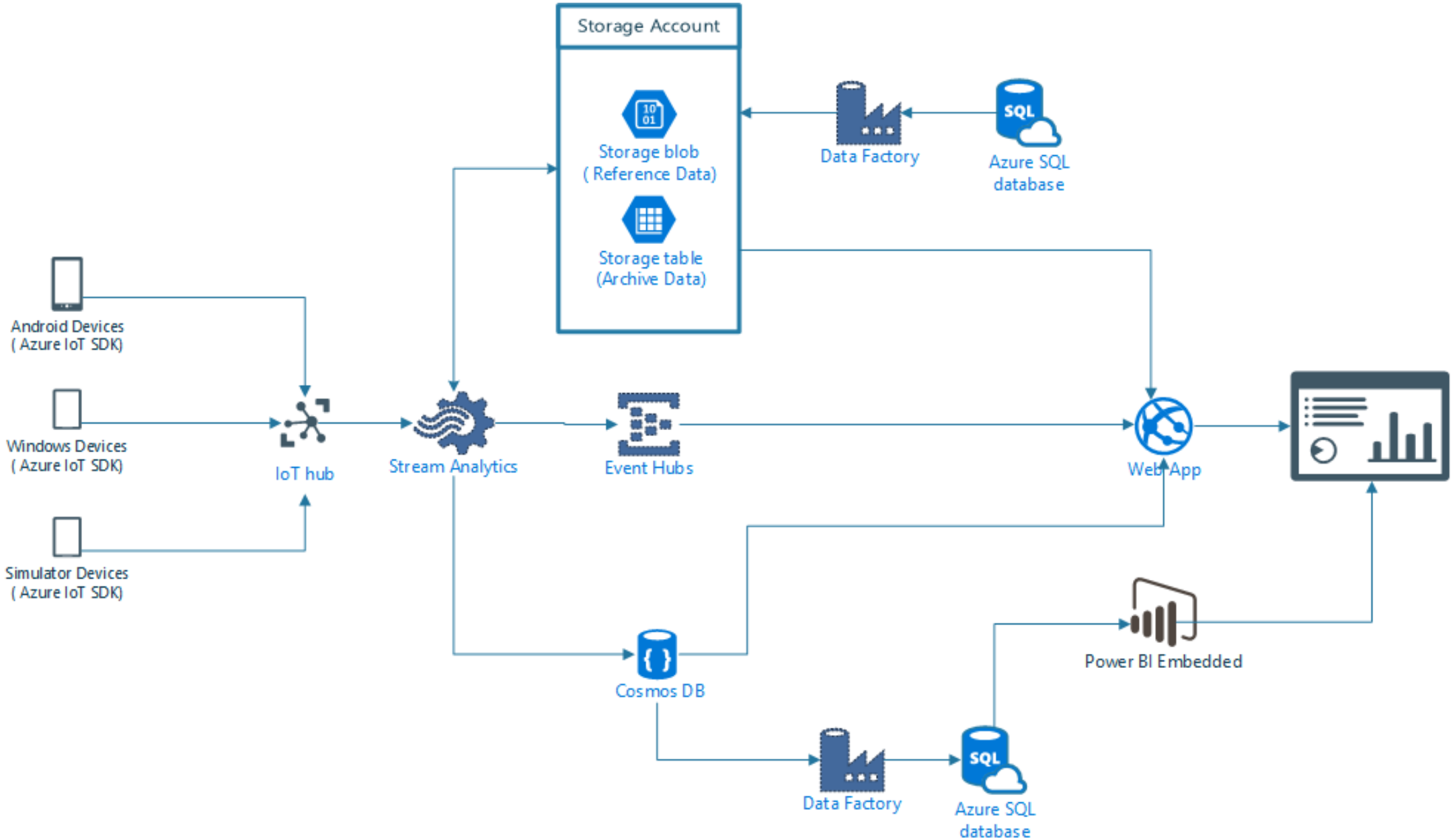
What to use for Data

Azure HDInsight

- When you need a **low cost, high throughput** data store.
- When you need to store **No-SQL** data.
- Provides a Hadoop **Platform as a Service** approach
- Suits acting as a **Hadoop, Hbase, Storm or Kafka** data store.
- **Eases the deployment and management** of clusters.

Azure Data Catalog

- When you require **documentation** of your data stores.
- When you require a **multi user** approach to documentation.
- When you need to **annotate data sources** with descriptive metadata.
- A **fully managed cloud service** whose users can discover the data sources.
- When you require **a solution that can help business users** understand their data.



https://www.blognone.com/node/93599

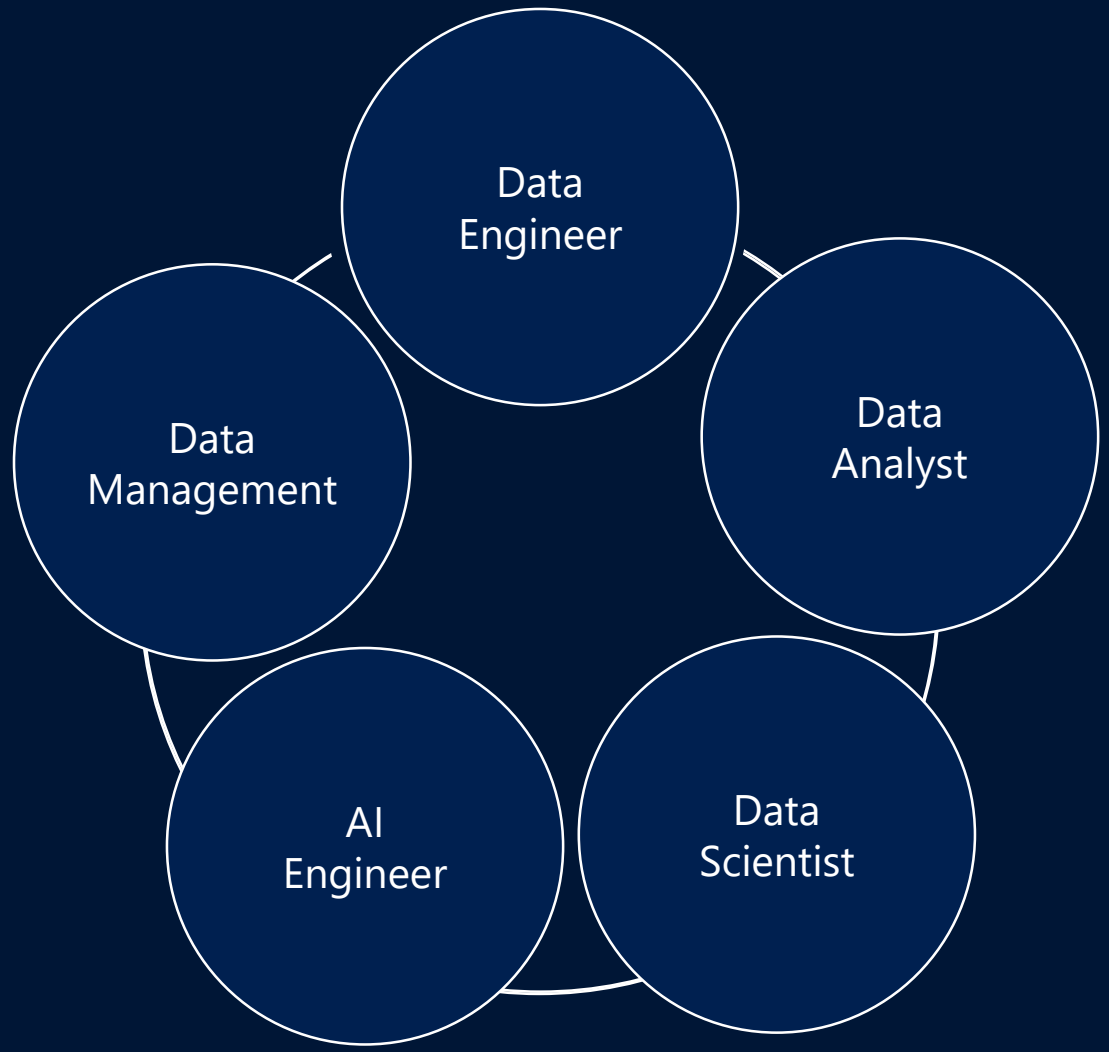
Lesson 03

Identify the Tasks Performed by a Data Engineer

Lesson Objectives

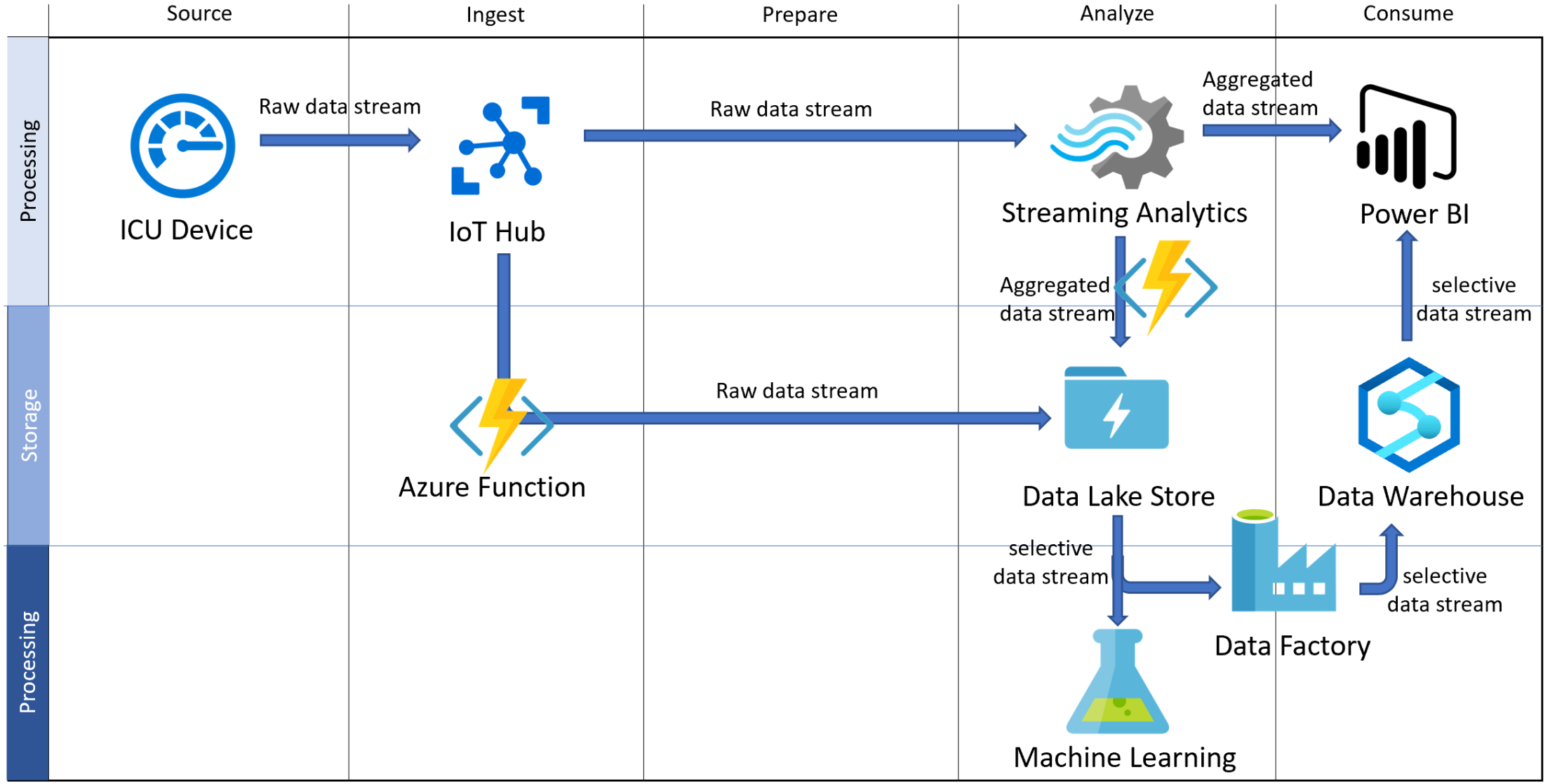
- List the new roles of modern data projects
- Outline data engineering practices
- Explore the high-level process for architecting a data engineering project

Roles in Data Projects



Data Engineering Practices

Architecting Projects – an example



Lesson 04

Course Case Study

Lesson Objectives

- Read the course case study

Course case study: AdventureWorks Cycles

Read the case study

In this section of the course, the instructor will either:

- Allocate you 10 minutes to read through the case study.

  Or

- Spend 10 minutes walking through the case study with you as a group

Note:
This case study will be used in labs across the entire course. Each lab will drill down more into the detail of what is required as you perform each lab.

Lab: Azure for the Data Engineer

Module Summary

In this module, you have learned about:

- The evolving world of data.
- The services in the Azure Data Platform.
- The tasks that are performed by a Data Engineer.
- A fictitious Case Study for use in labs.

Next steps

After the course, consider visiting [the Microsoft Customer Case Study site. Use the search bar to search by an industry such as healthcare or retail, or by a technology such as Azure Cosmos DB or Stream Analytics. Read through some of the customers stories.

Microsoft

Module 02: Working with Data Storage

Agenda

- L01 – Choose a data storage approach in Azure
- L02 - Create an Azure Storage Account
- L03 - Explain Azure Data Lake Storage
- L04 - Upload data into Azure Data Lake Store

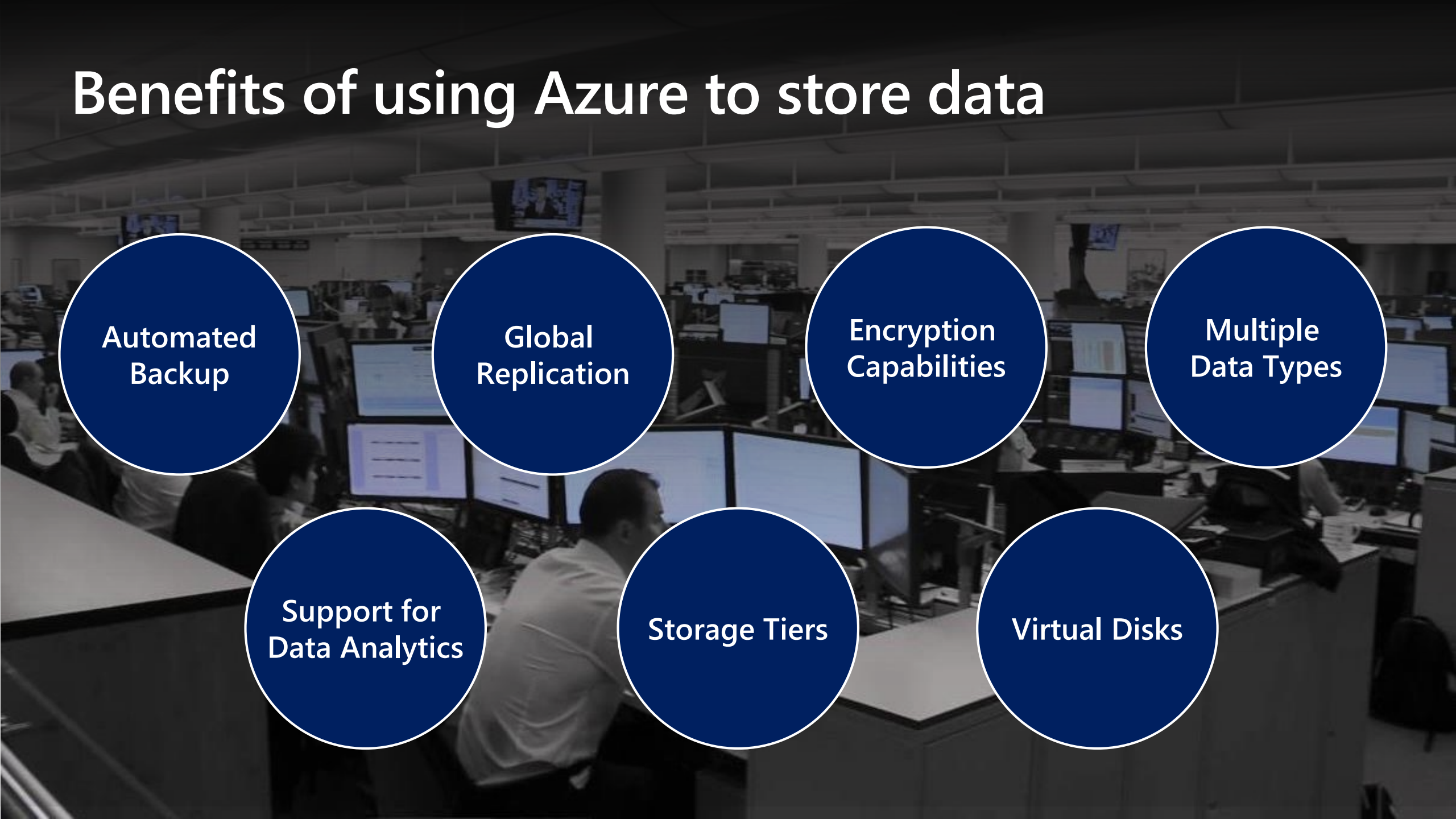
Lesson 01

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



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 an 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.

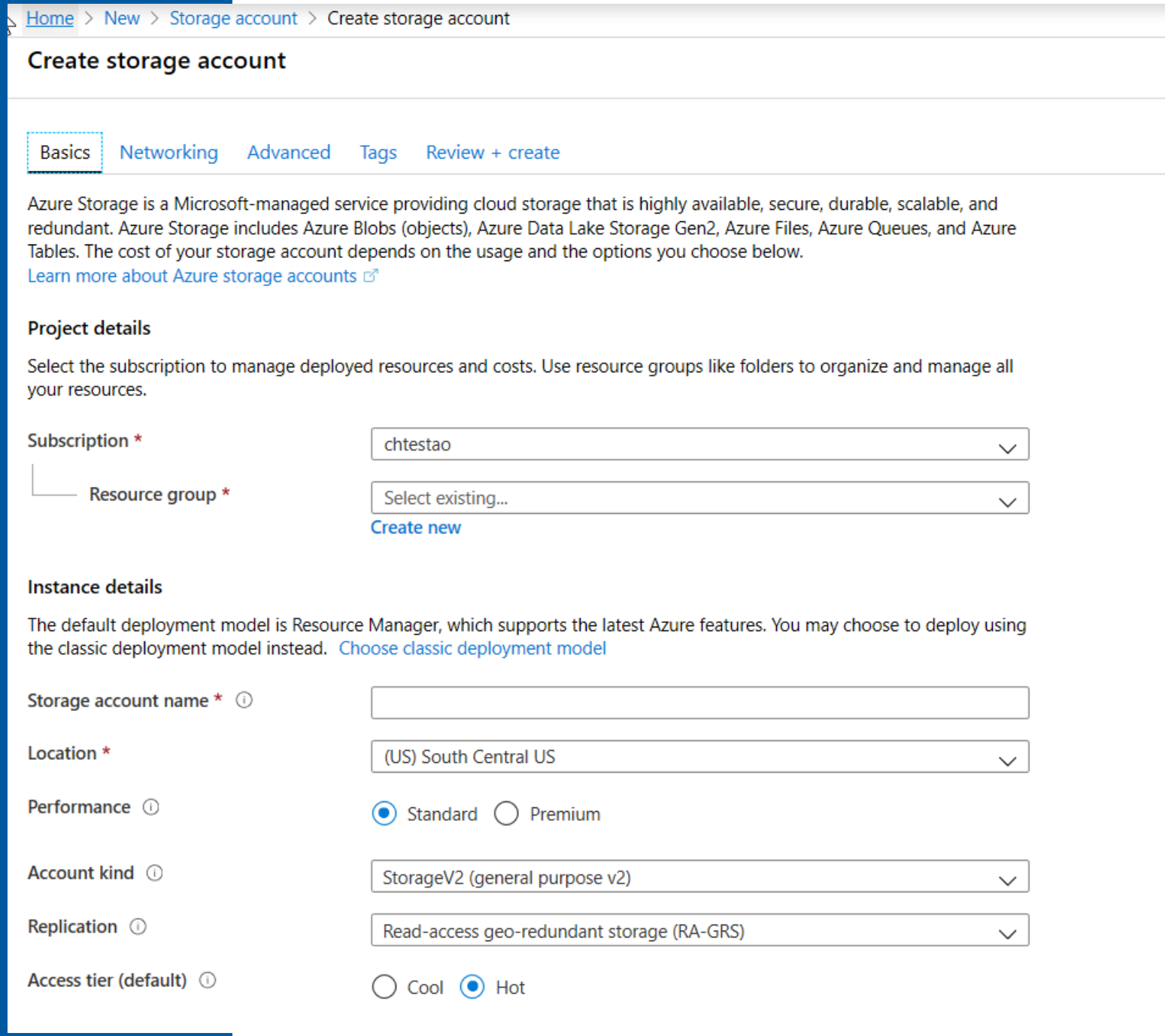
Cost Sensitivity

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

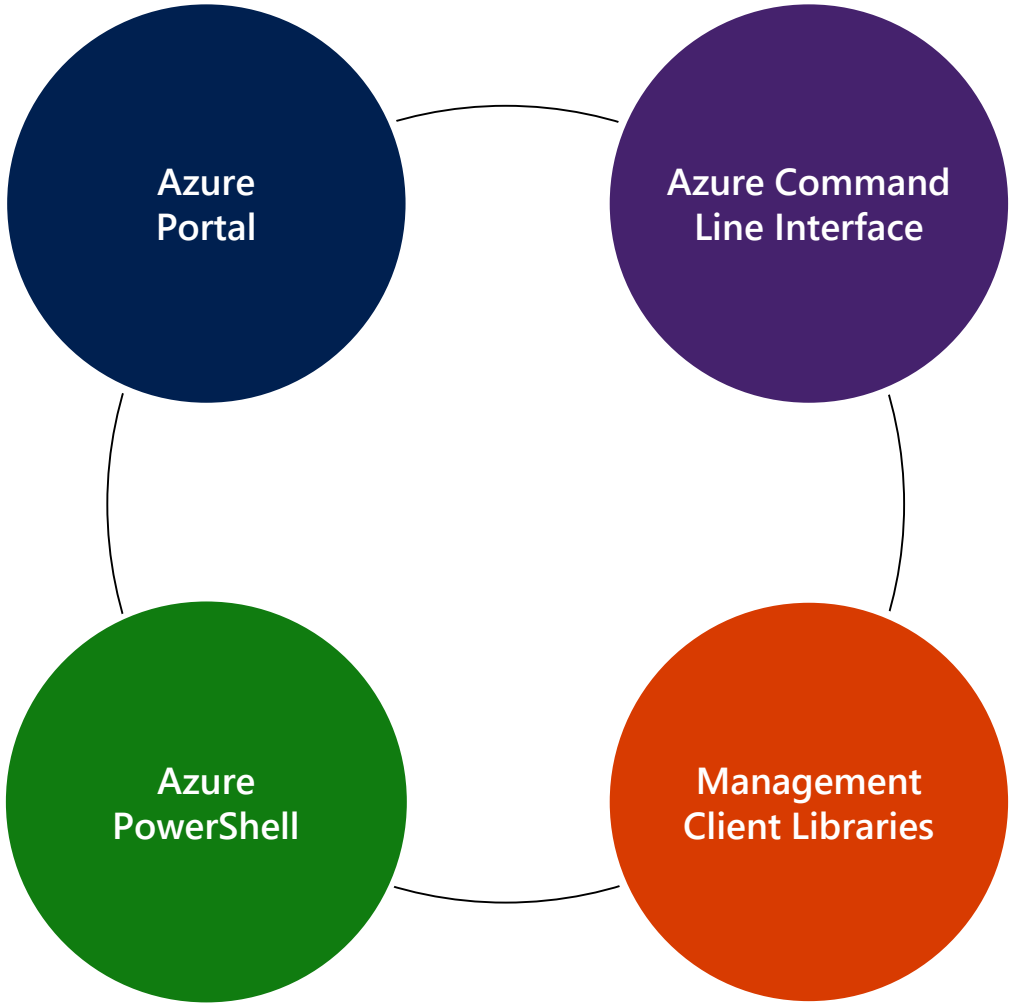
Management Overhead

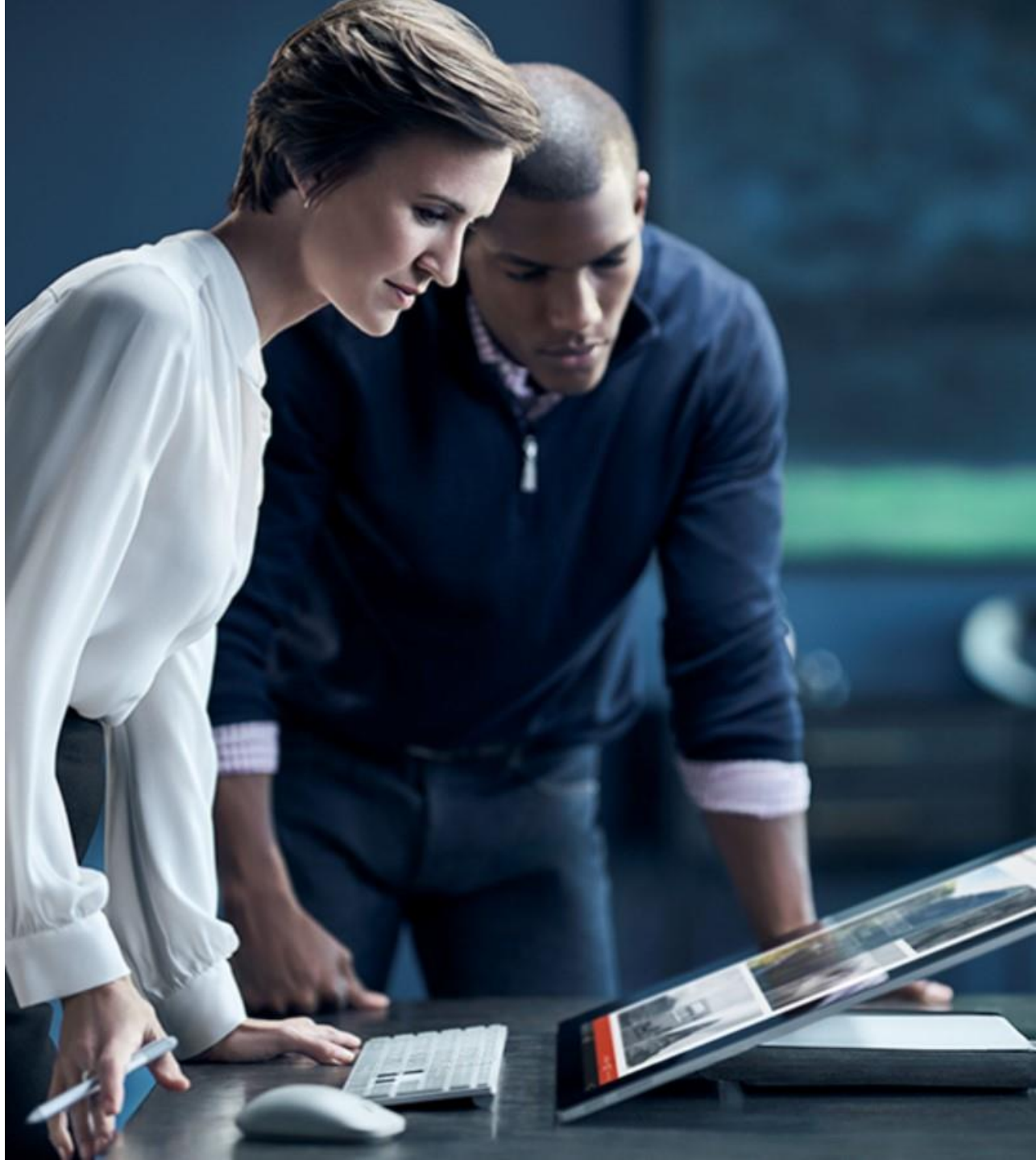
Each storage account requires some time and attention from an administrator to create and maintain.

Storage Account settings



Storage Account creation tool





Create a Storage Account

Lesson 03

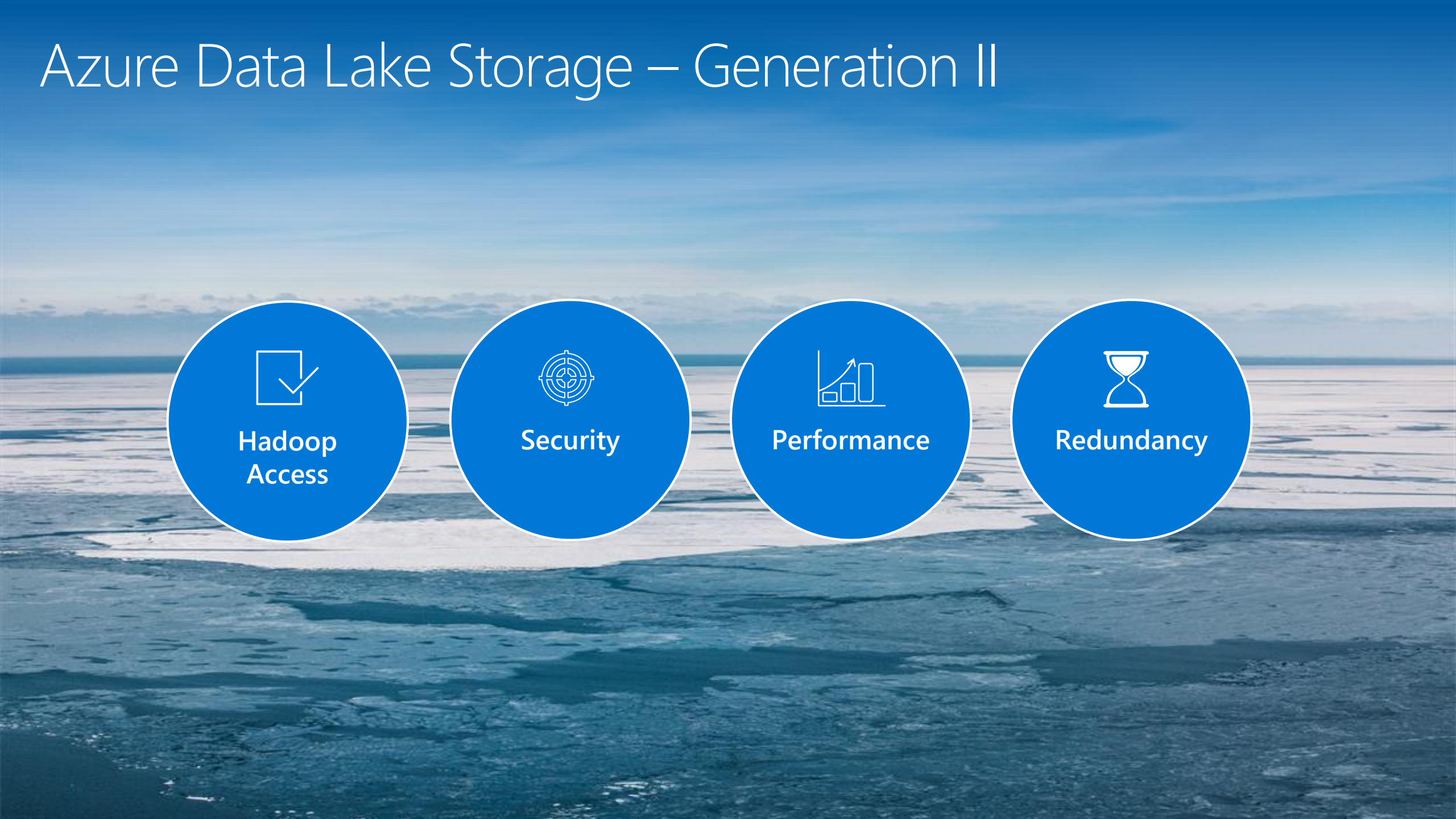
Azure Data Lake Storage

Start = 13.00

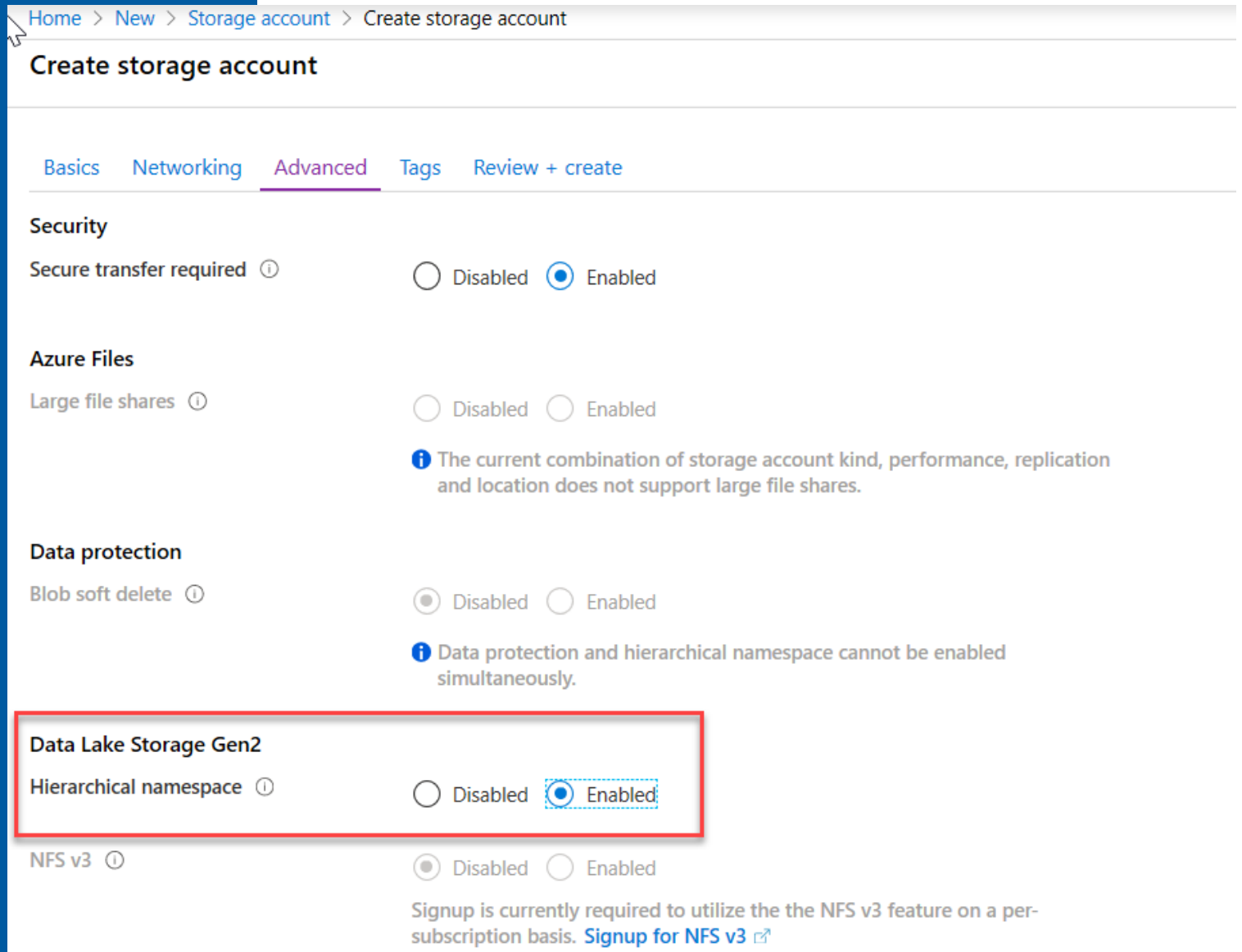
Lesson Objectives

- 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

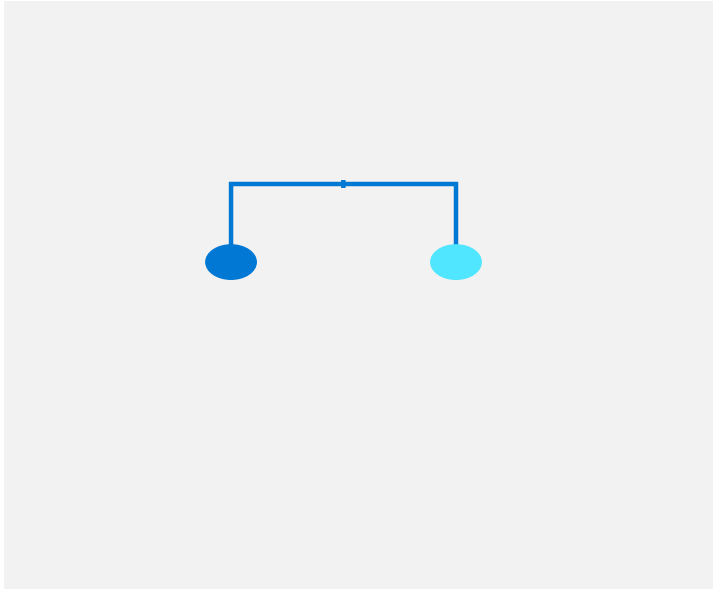


Create a Azure Data Lake Store (Gen II) using the Portal.

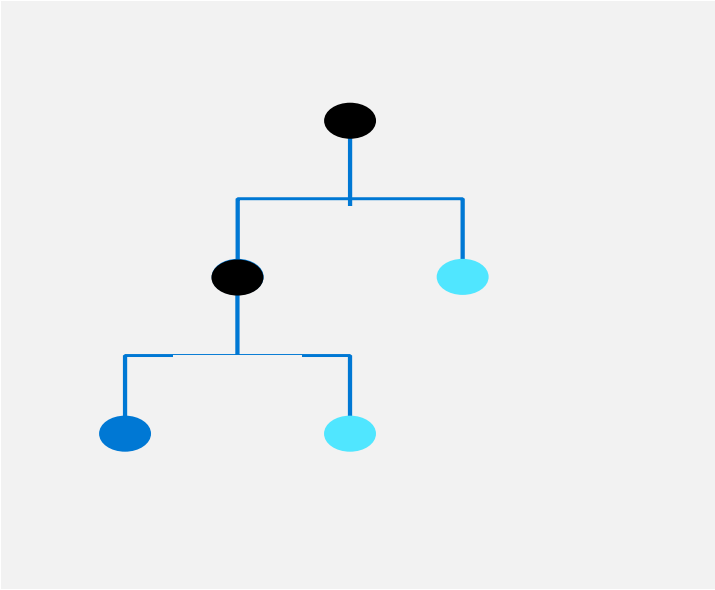


Compare Azure Blob Storage and Data Lake Store Gen 2

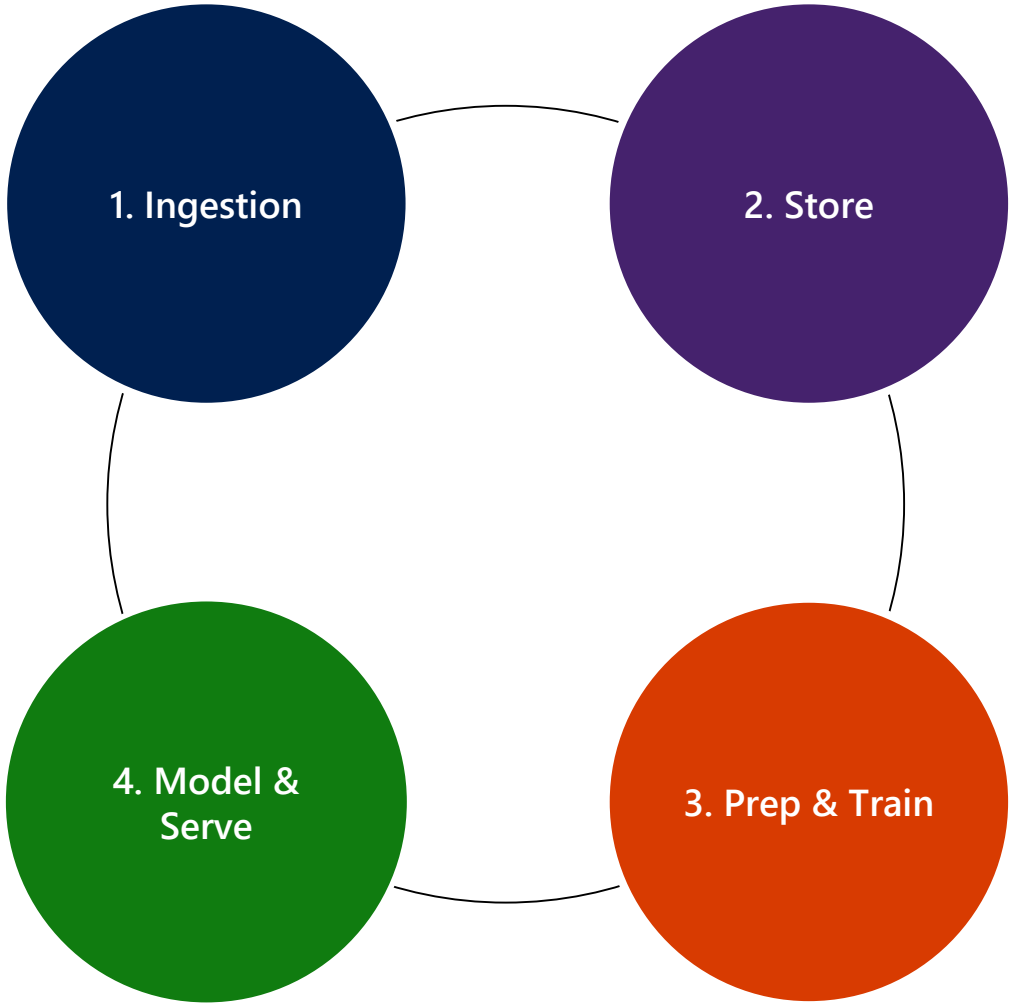
Azure Blob
Flat Namespace

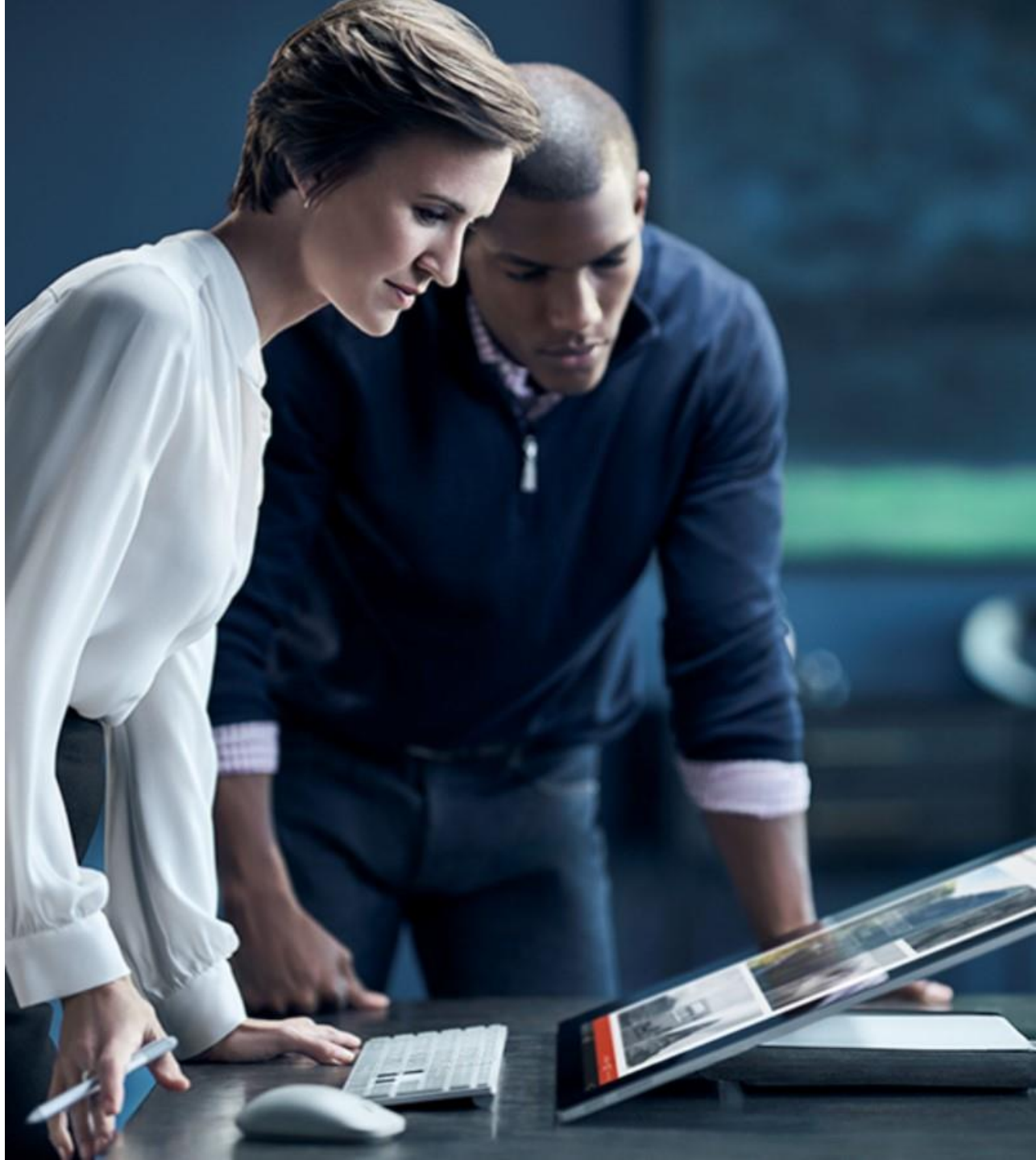


Data Lake (Gen II)
Hierarchical Namespace



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.

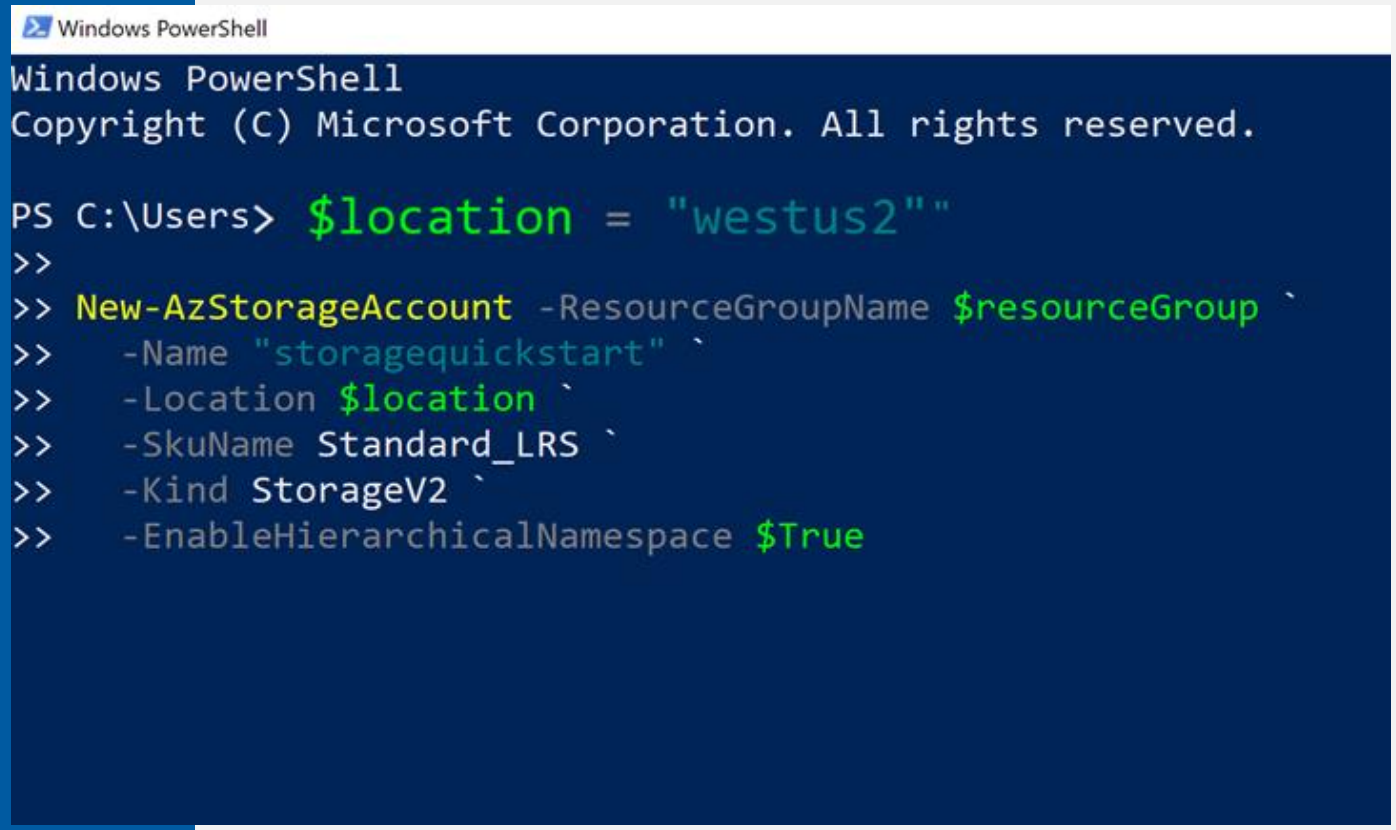
Lesson 04

Upload Data into Azure Data Lake Store

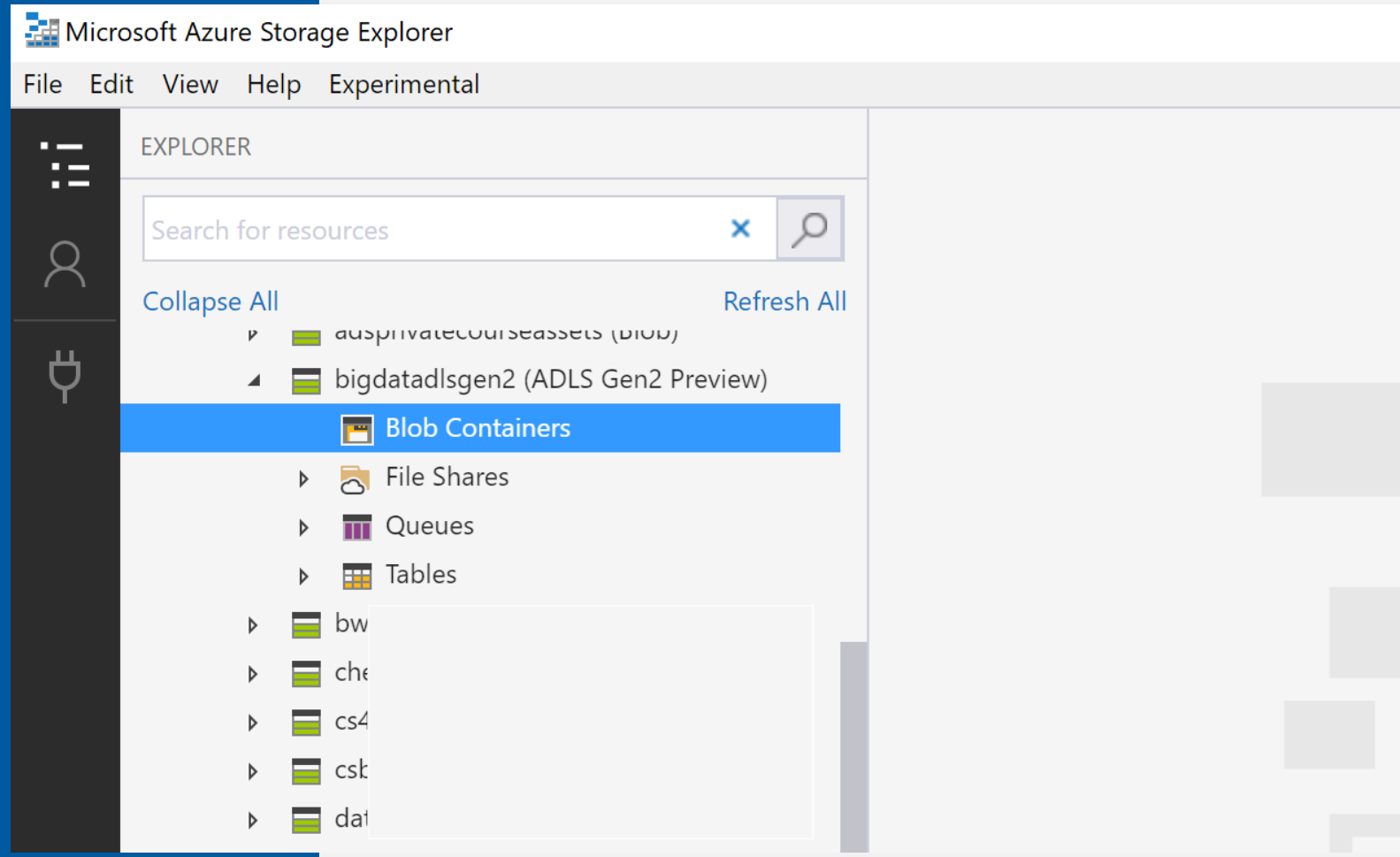
Lesson Objectives

- 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

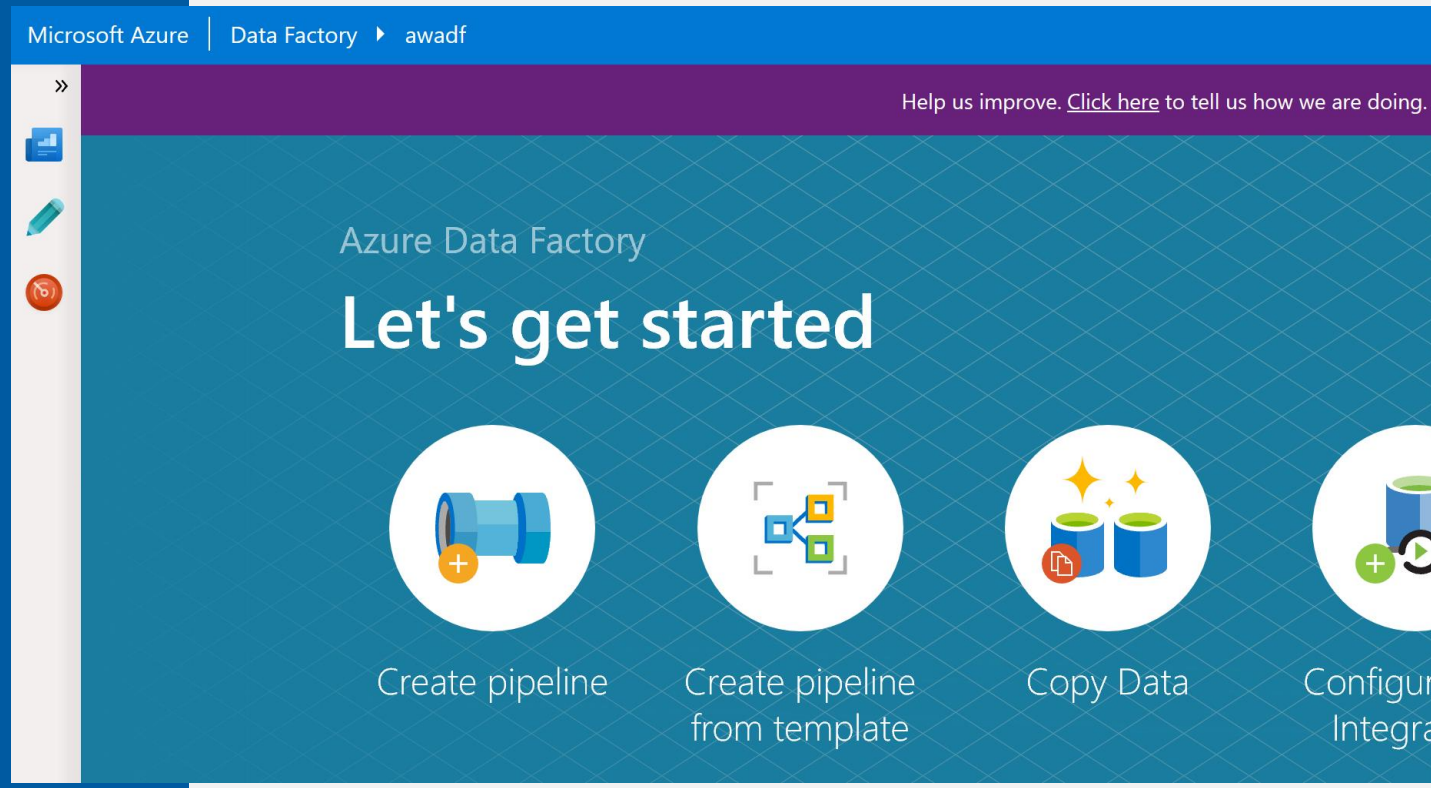
Create a Azure Data Lake Store (Gen II) using PowerShell.



Uploading data with Azure Storage Explorer



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



Lab: Working with Data Storage

Microsoft

Module 03: Enabling Team Based Data Science with Azure Databricks

Agenda

- L01 – Describe Azure Databricks
- L02 - Provision Azure Databricks and Workspaces
- L03 - Read data using Azure Databricks
- L04 - Perform transformations with Azure Databricks

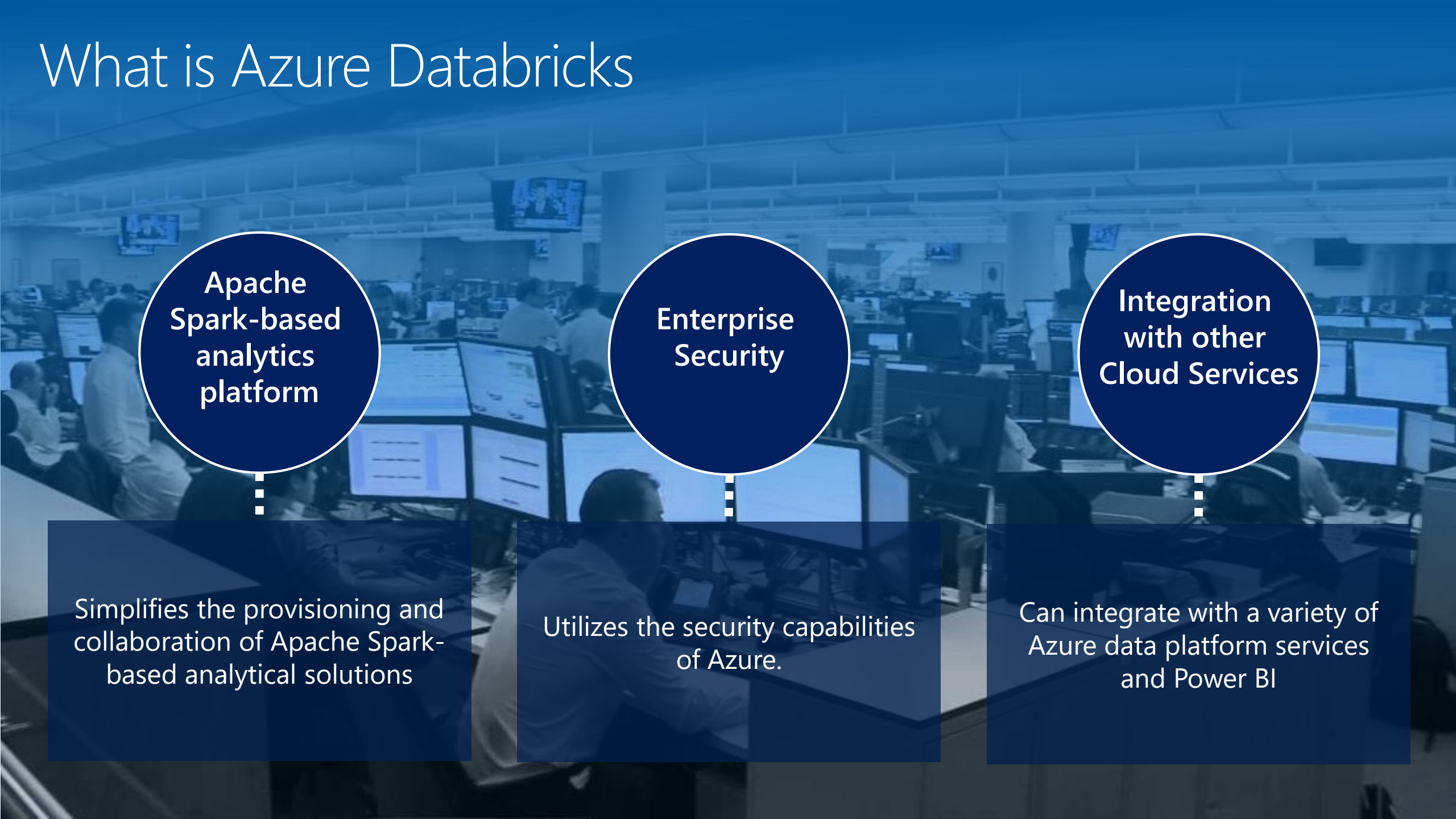
Lesson 01

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

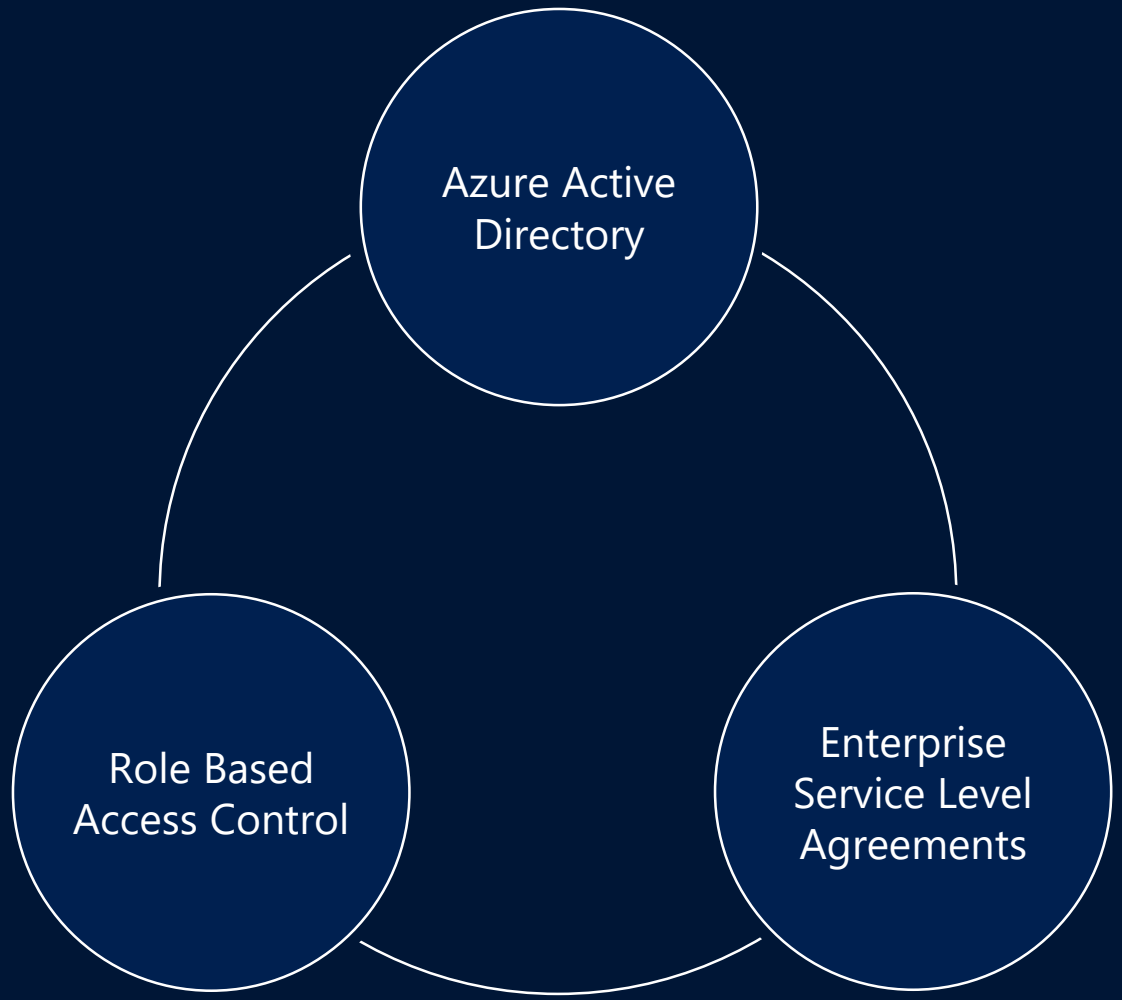
Why use Azure Databricks

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

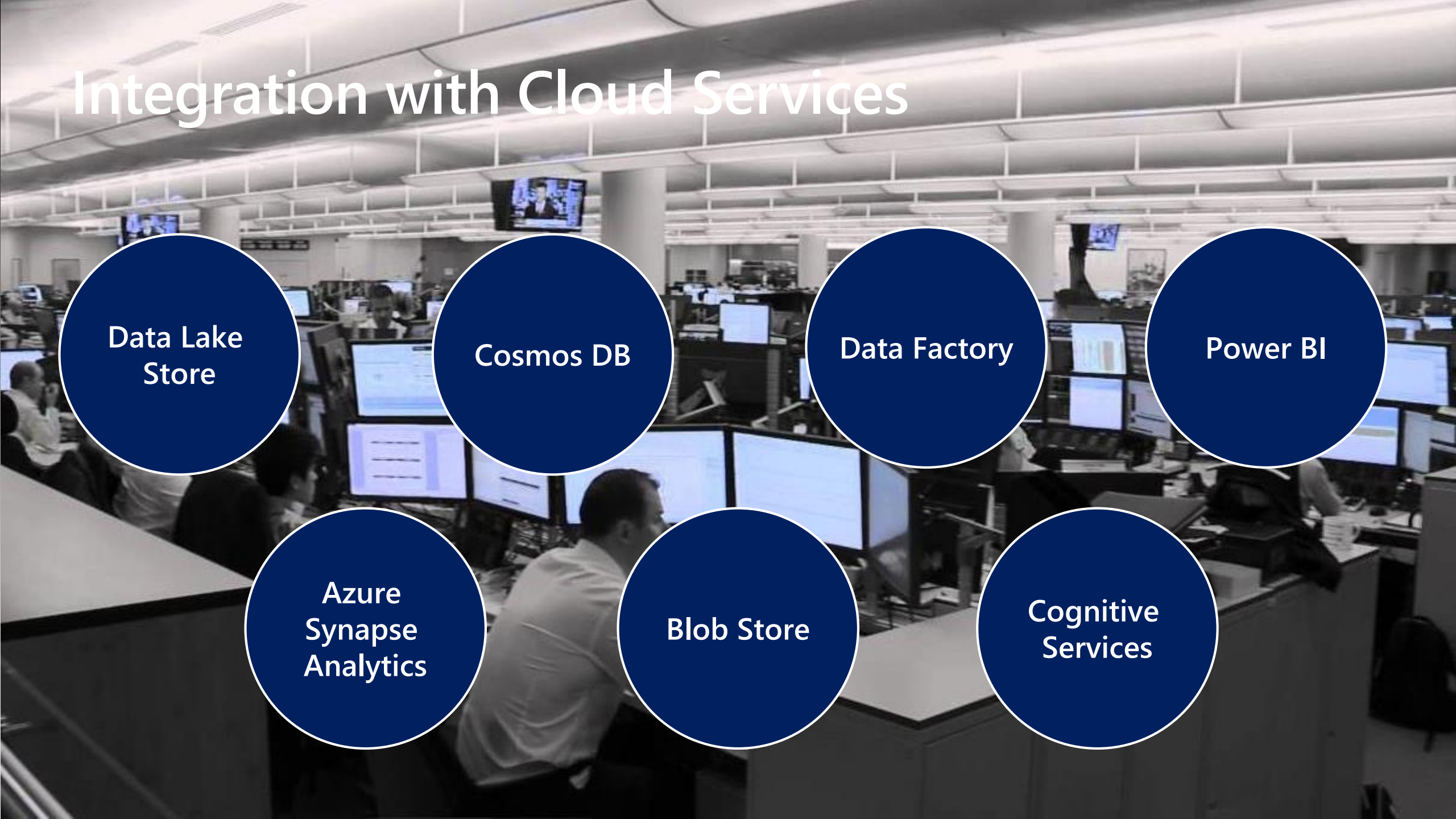
Azure Databricks components:

- Spark SQL and DataFrames
- Streaming
- Mlib
- GraphX
- Spark Core API

Enterprise Security



Integration with Cloud Services



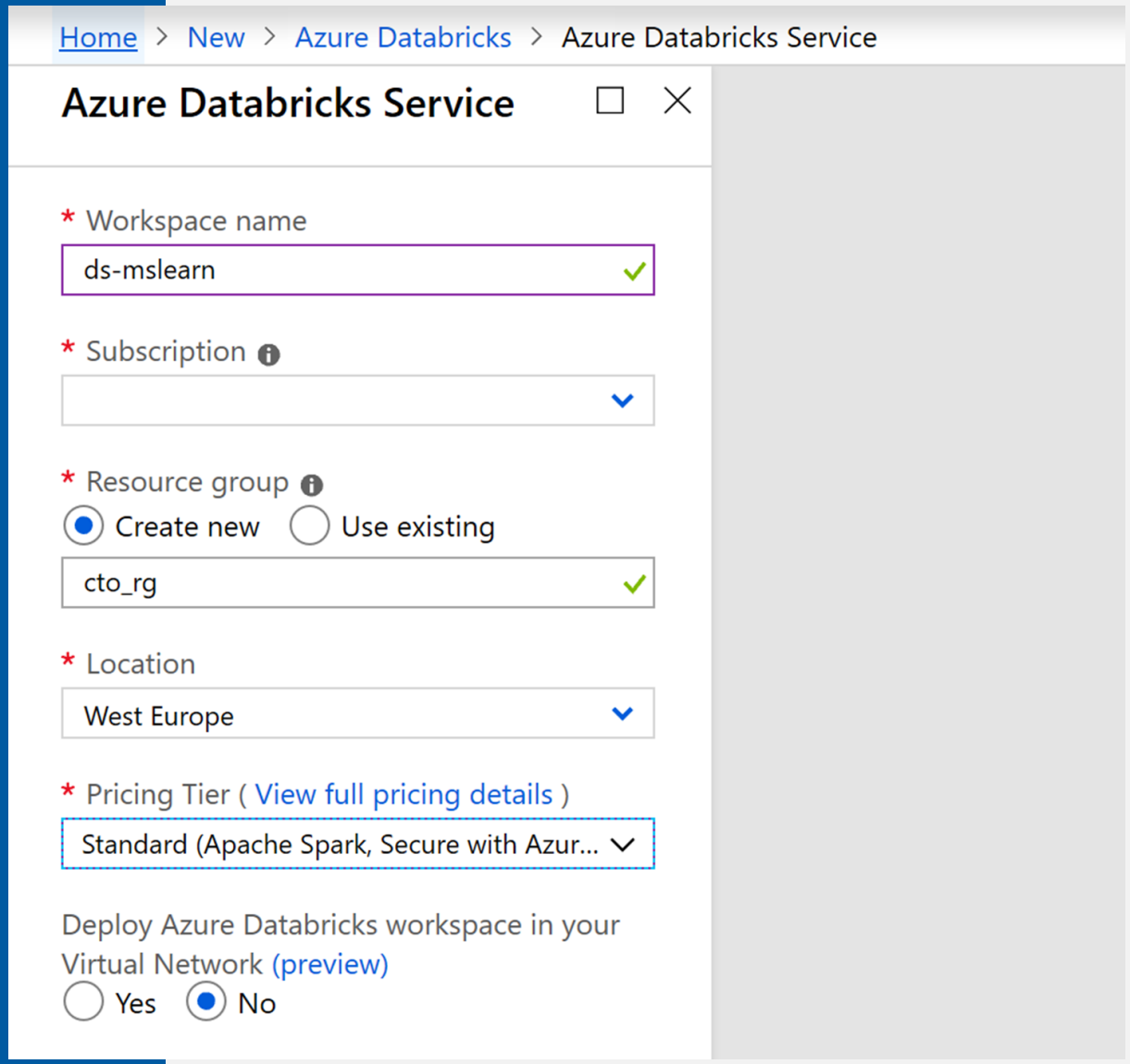
Lesson 02

Provision Azure Databricks and Workspaces

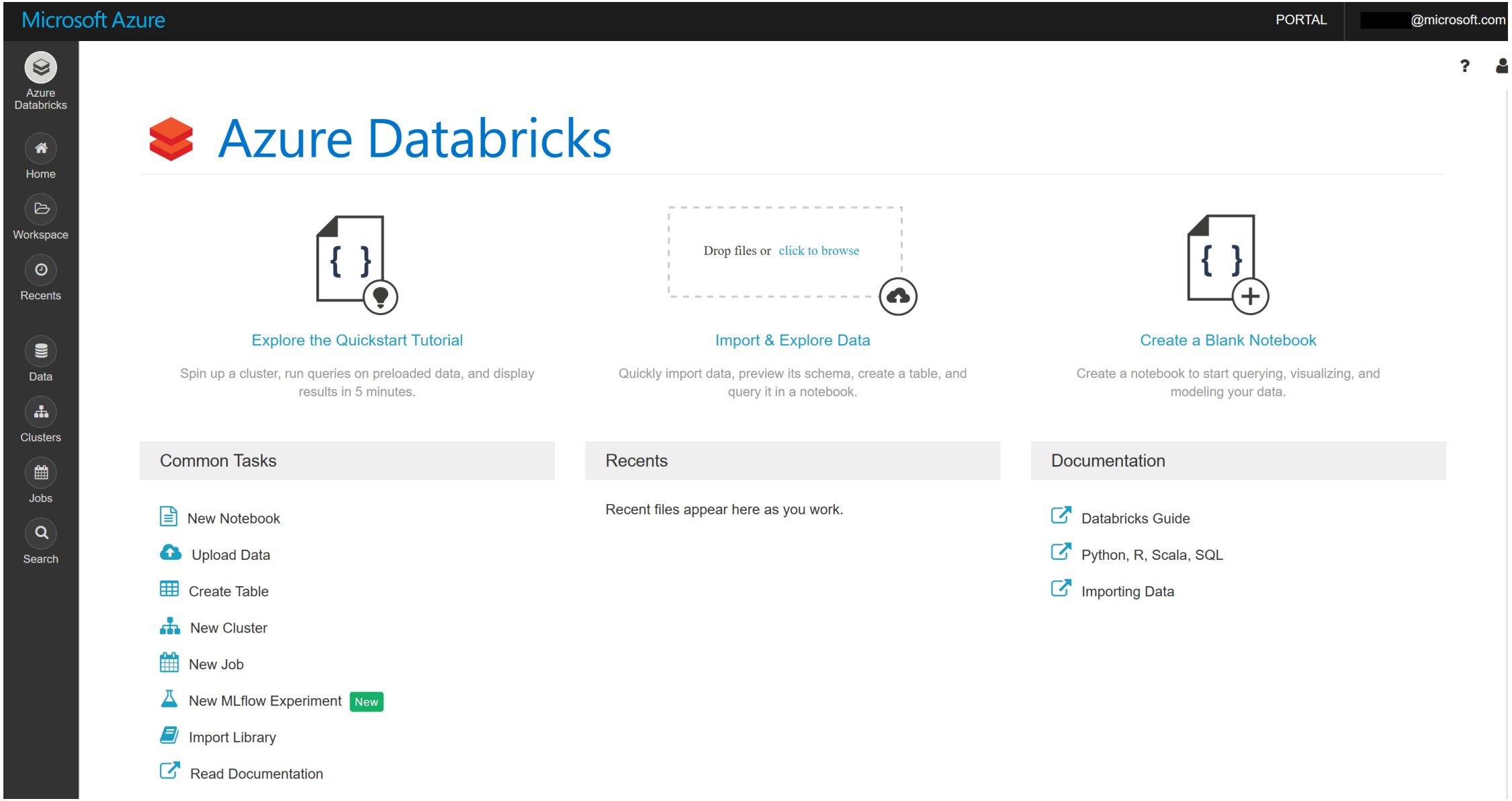
Lesson Objectives

- Create your own Azure Databricks workspace
- Create a cluster and notebook in Azure Databricks

Create an Azure Databricks Workspace.



Create a Cluster and Notebook in Azure Databricks



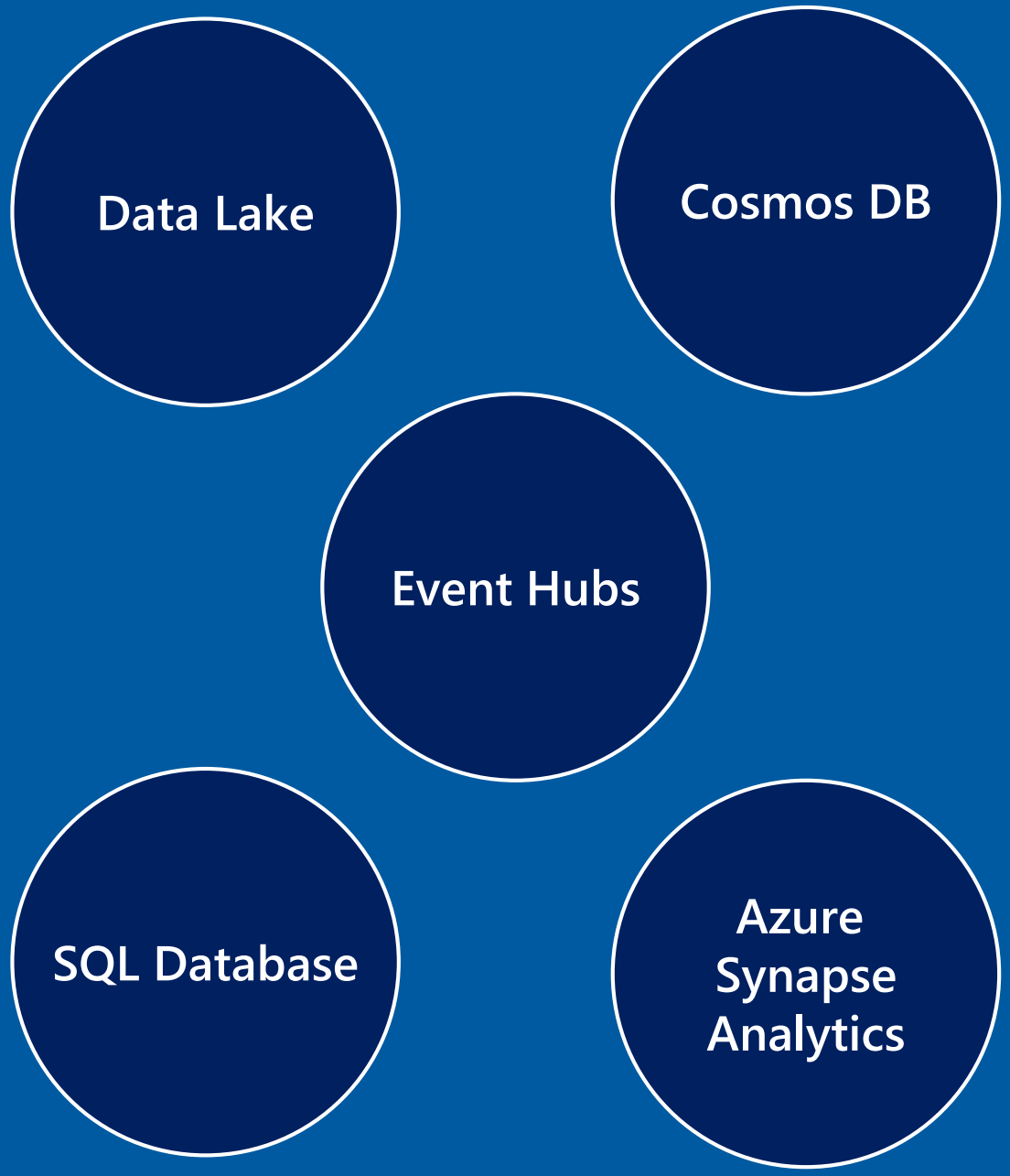
Lesson 03

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
Connect to many data stores: • 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

Microsoft

Module 04: Building Globally Distributed Databases with Cosmos DB

Agenda

- L01 – Create an Azure Cosmos DB database built to scale
- L02 - Insert and query data in your Azure Cosmos DB database
- L03 - Build a .NET Core app for Azure Cosmos DB in Visual Studio Code
- L04 - Distribute your data globally with Azure Cosmos DB

Lesson 01

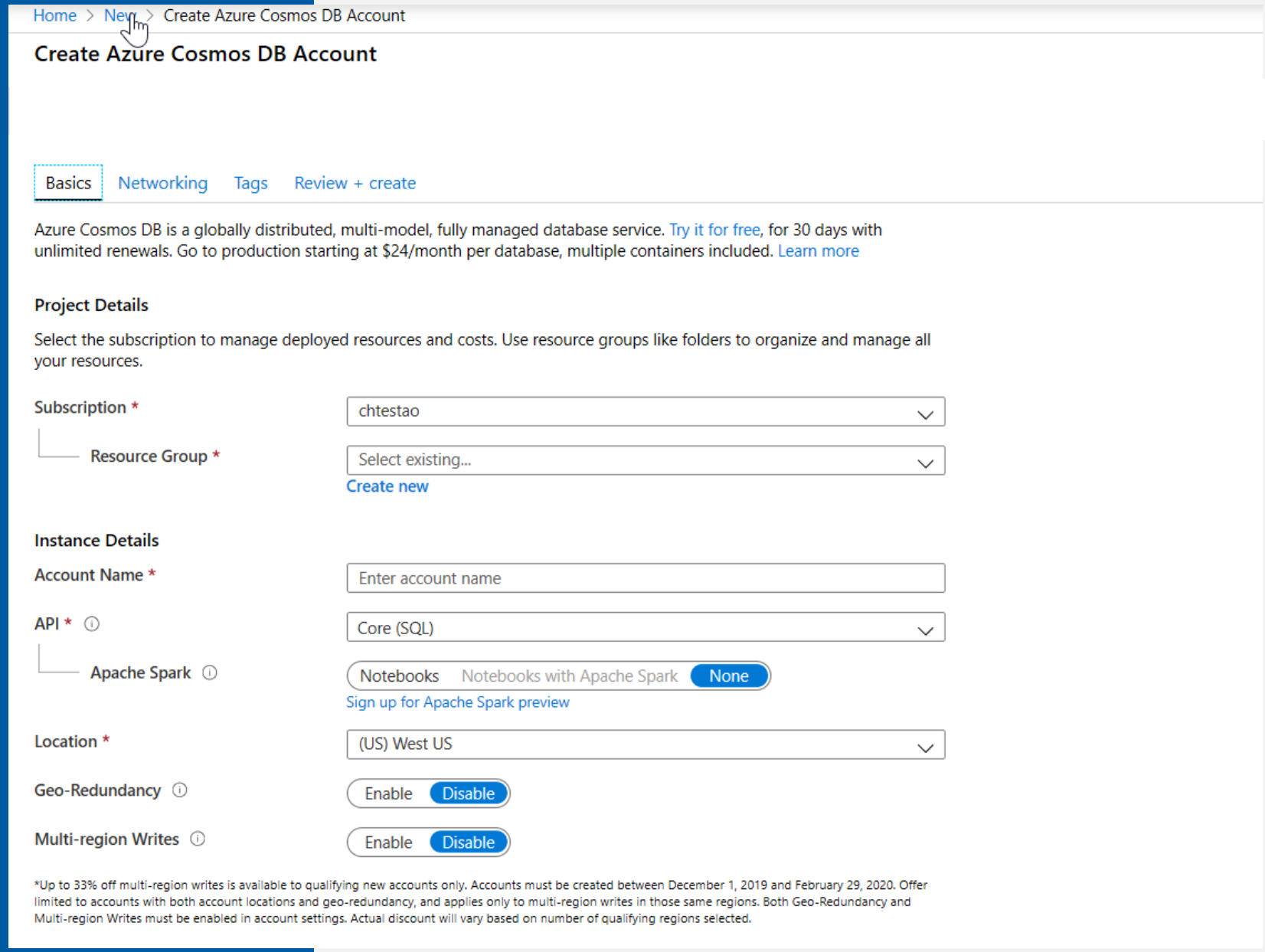
Create an Azure Cosmos DB Database Built to Scale

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

Create an Azure Cosmos DB account.



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

Request unit basics

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

Best Practice.

Range of Values

The more values your partition key has, the more scalability you have

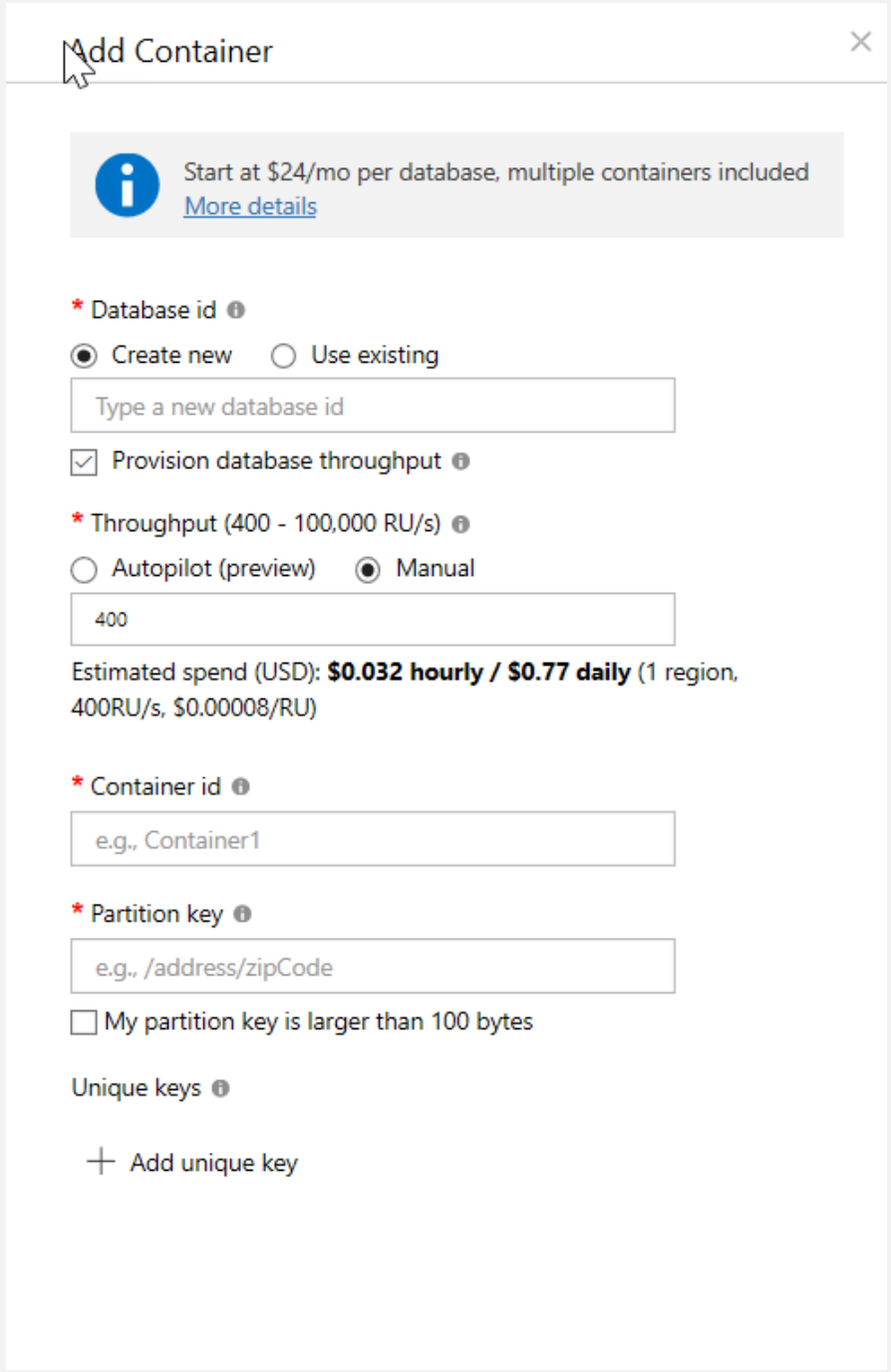
Review Queries

To determine the best partition key for a read-heavy workload, review the top three to five queries you plan on using

Transactional Workloads

For write-heavy workloads, you'll need to understand the transactional needs of your workload

Creating a Database and a Container in Cosmos DB



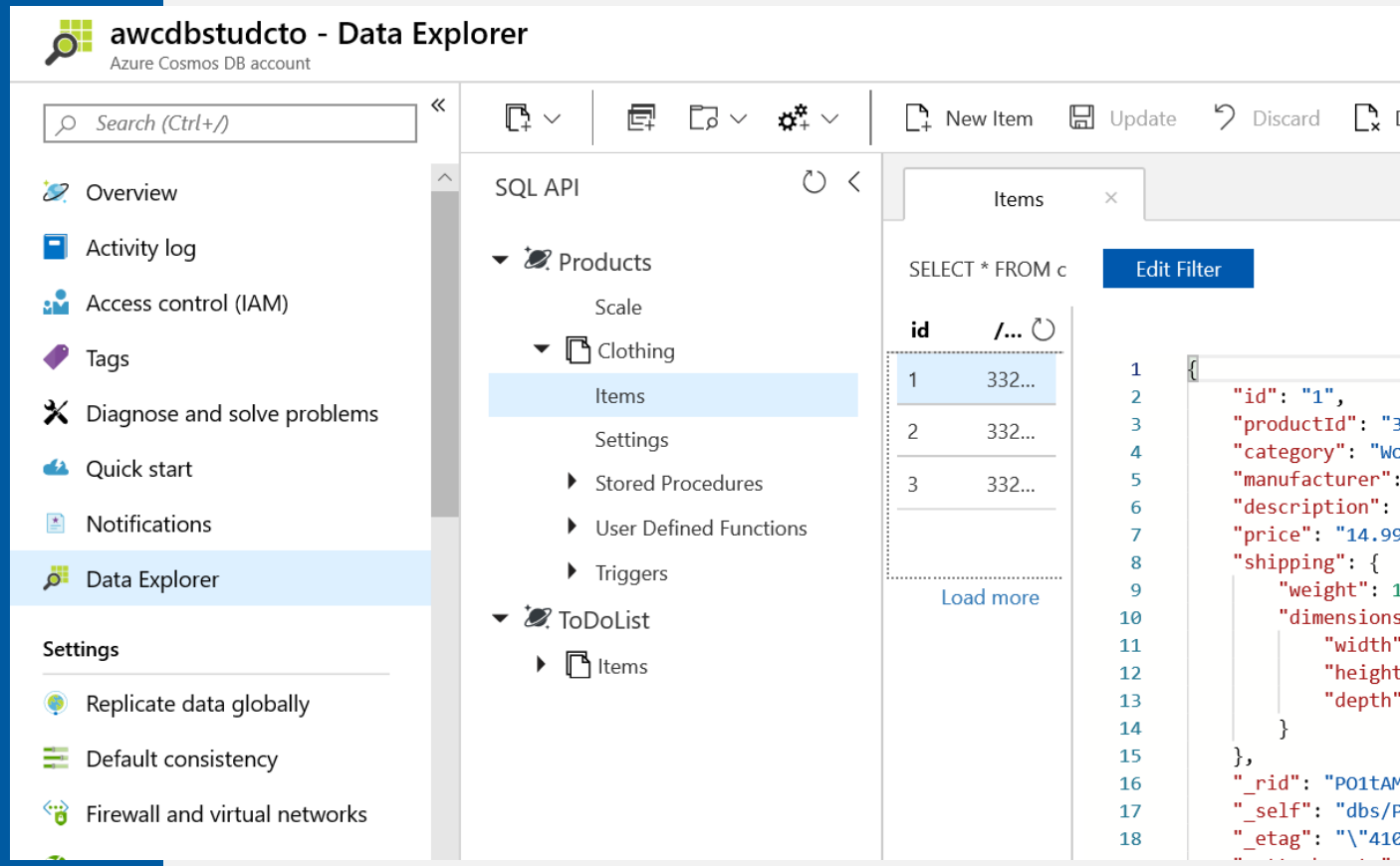
Lesson 02

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.



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'))"
]
```

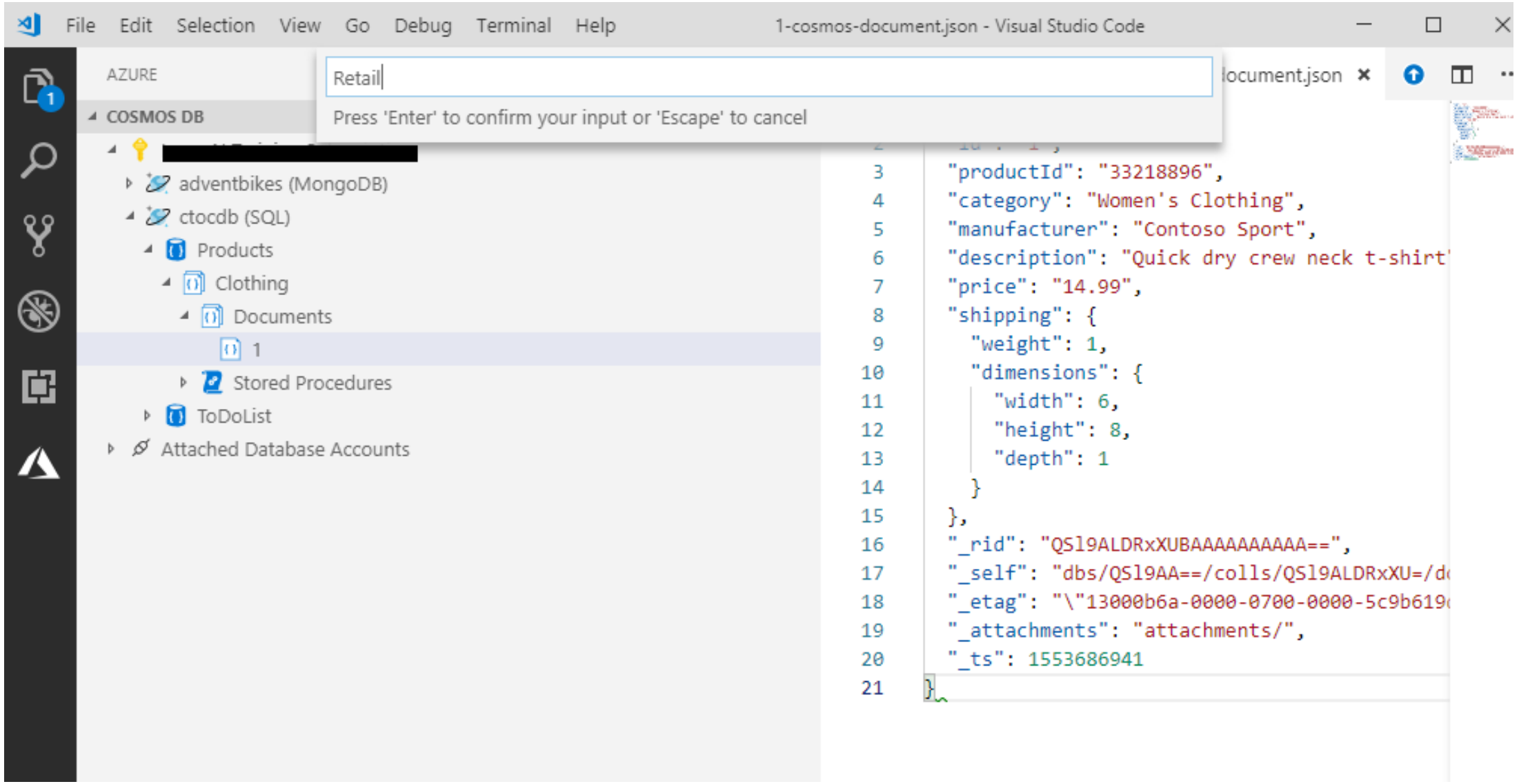
Lesson 03

Build a .NET Core App for Azure Cosmos DB in VS Code

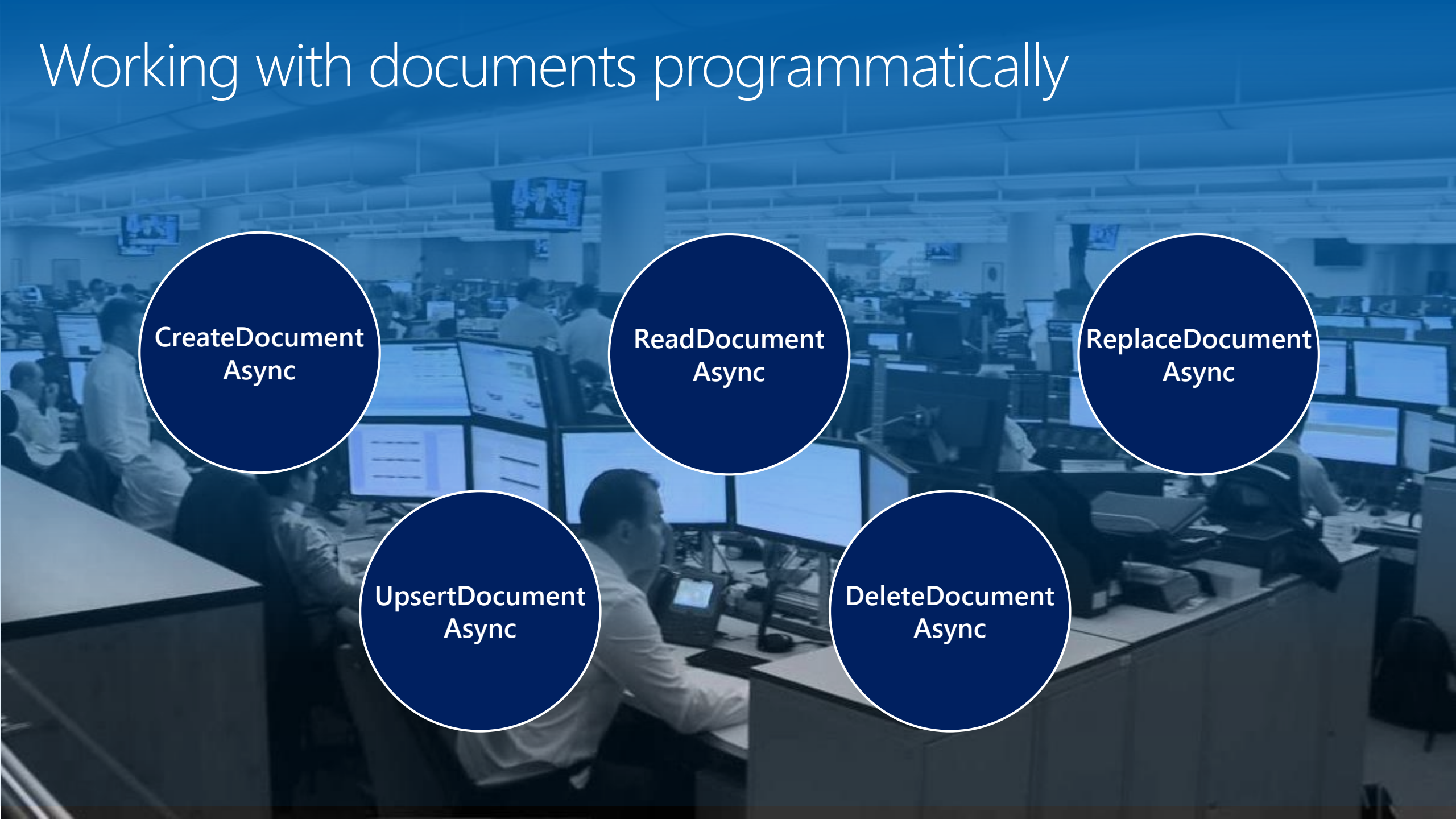
Lesson Objectives

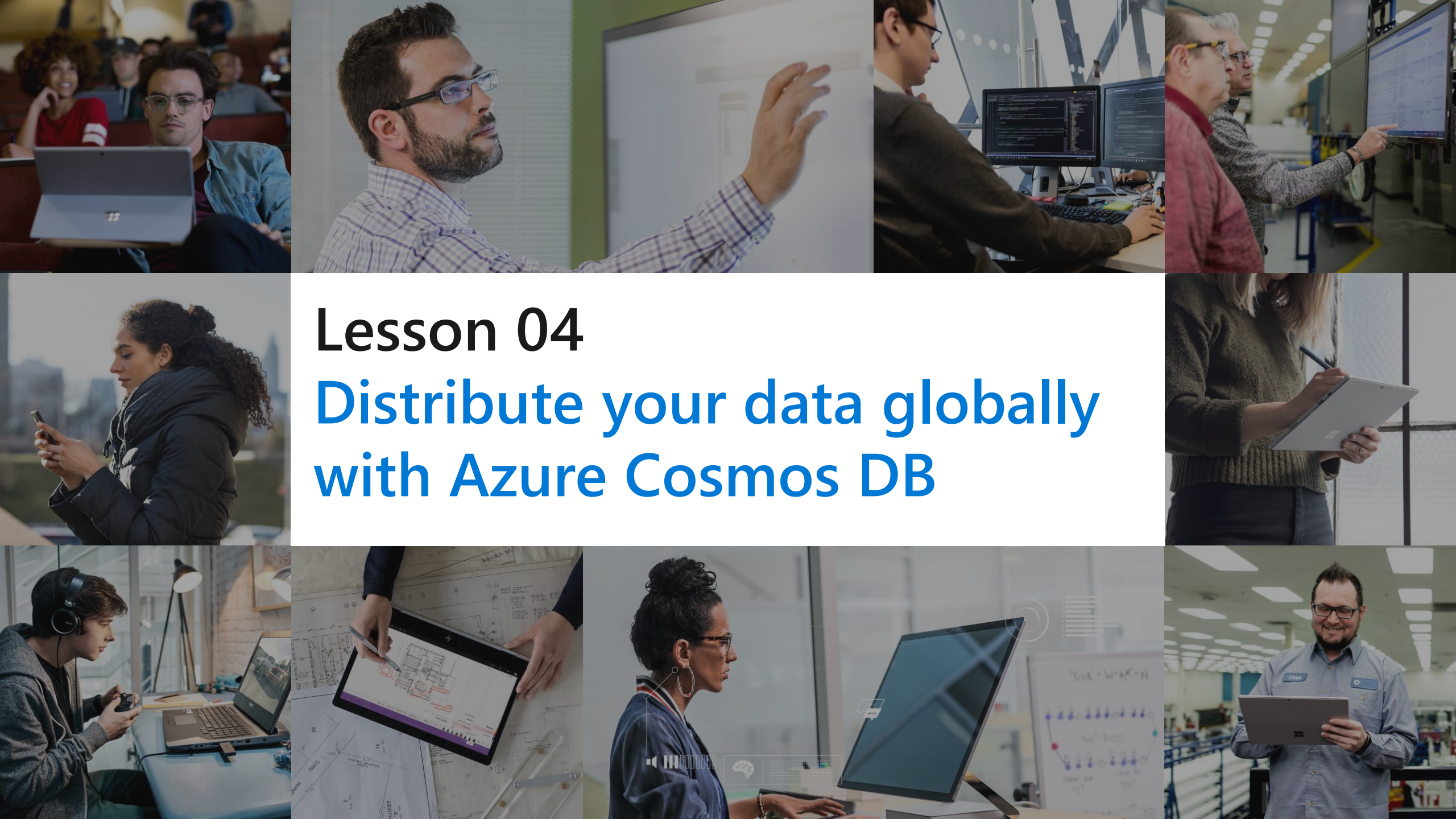
- Create an Azure Cosmos DB account, database, and container in Visual Studio Code using the Azure Cosmos DB extension
- Create an application to store and query data in Azure Cosmos DB
- Use the Terminal in Visual Studio Code to quickly create a console application
- Add Azure Cosmos DB functionality with the help of the Azure Cosmos DB extension for Visual Studio Code

Creating Azure Cosmos DB in Visual Studio Code



Working with documents programmatically





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 > ctocdb > Replicate data globally

Replicate data globally

ctocdb

Save | Discard | Manual Failover | Automatic Failover

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

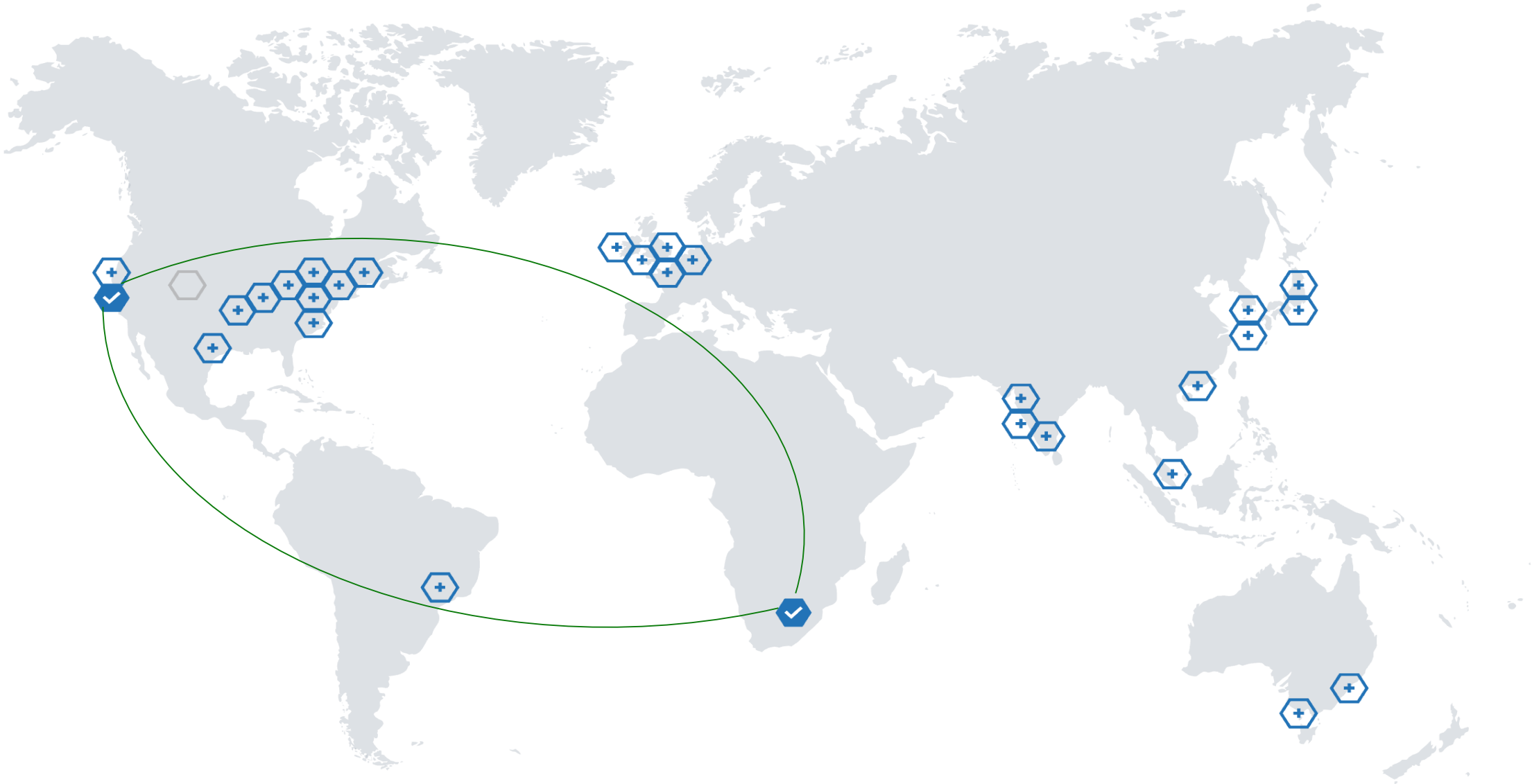
* Each region is billable based on the throughput and storage for the account. Learn more

Configure regions

Configure the regions available for reads and writes. + Add region

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

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

Microsoft

Module 05: Working with Relational Data Stores in the Cloud

Start : 14.50 น.

Agenda

- L01 - Work with Azure SQL Database
- L02 - Work with Azure Synapse Analytics
- L03 - Provision and query data in Azure Synapse Analytics
- L04 - Import data into Azure Synapse Analytics using PolyBase

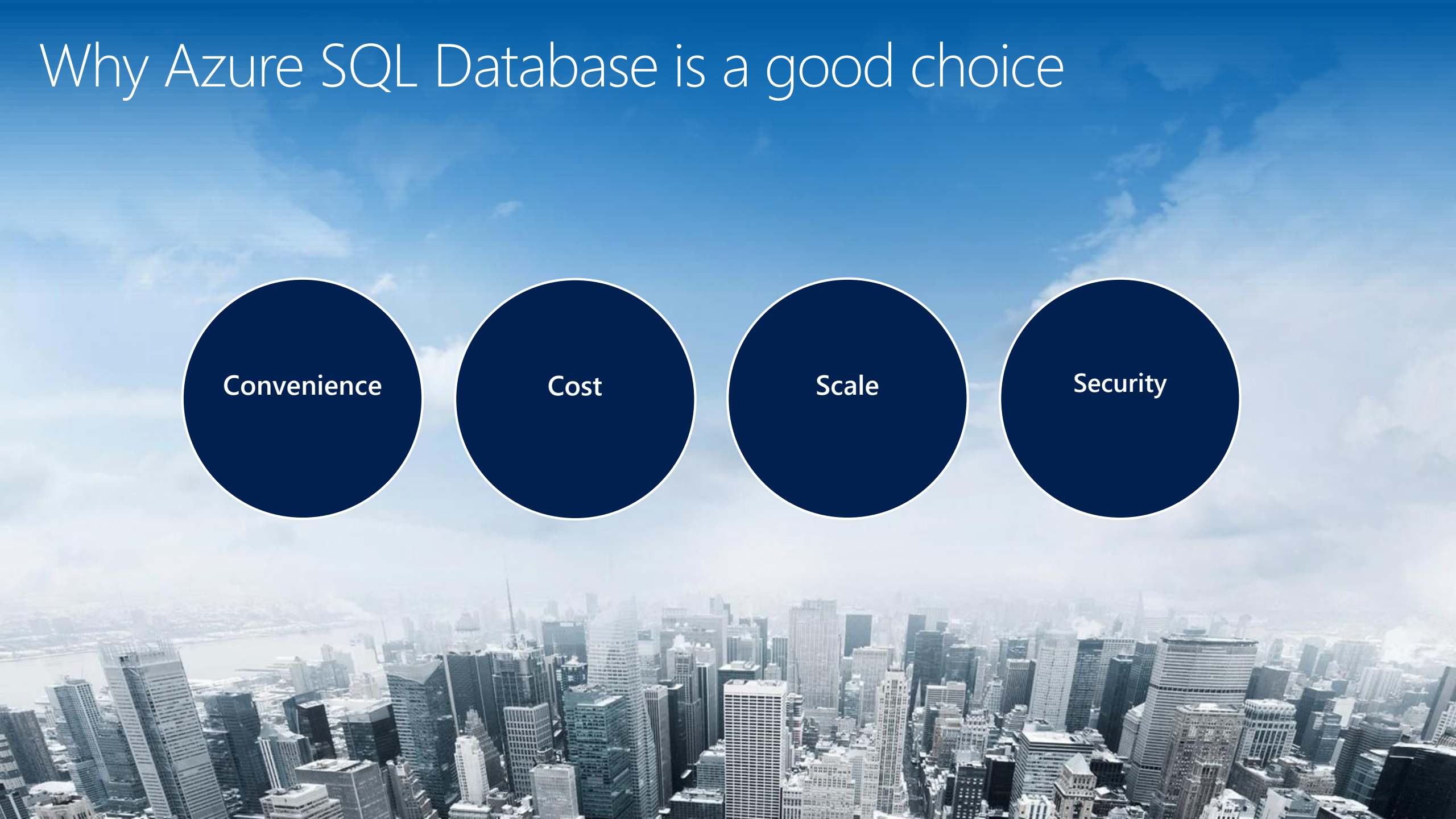
Lesson 01

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

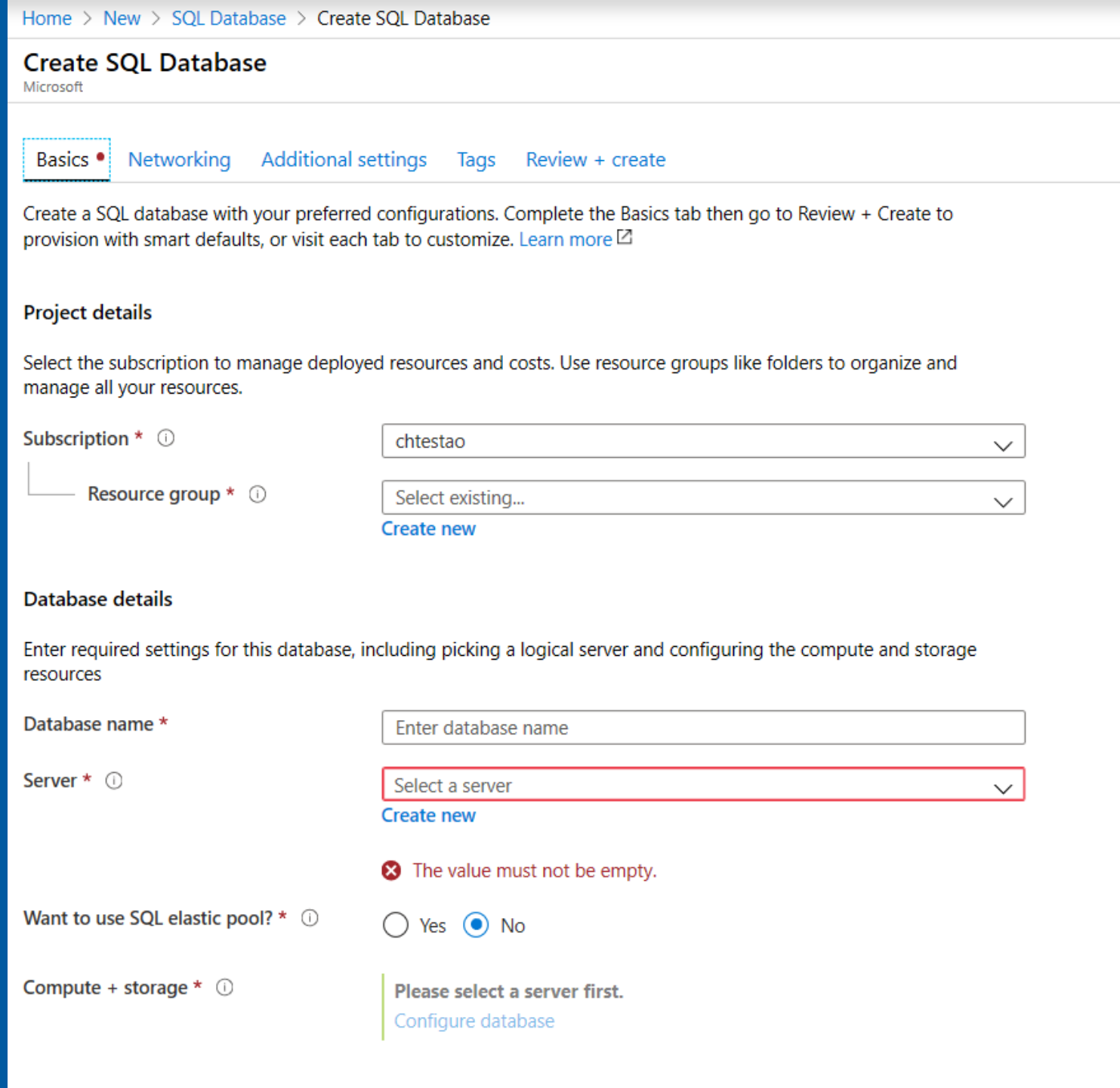


Azure SQL Database configuration options

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.



Use Azure Cloud Shell to connect to your Azure SQL database

```
sqlcmd -S tcp:contoso-1.database.windows.net,1433
-d Logistics -U martina -P "password1234$" -N -l 30
```

```
CREATE TABLE Drivers (DriverID int, LastName
varchar(255), FirstName varchar(255), OriginCity
varchar(255)); GO
```

```
SELECT name FROM sys.tables; GO
```

```
INSERT INTO Drivers (DriverID, LastName,
FirstName, OriginCity) VALUES (123, 'Zirne', 'Laura',
'Springfield'); GO
```

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

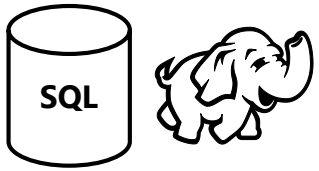
A materialized view pre-computes, stores, and maintains its data like a table. They are automatically updated when data in underlying tables are changed.

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"

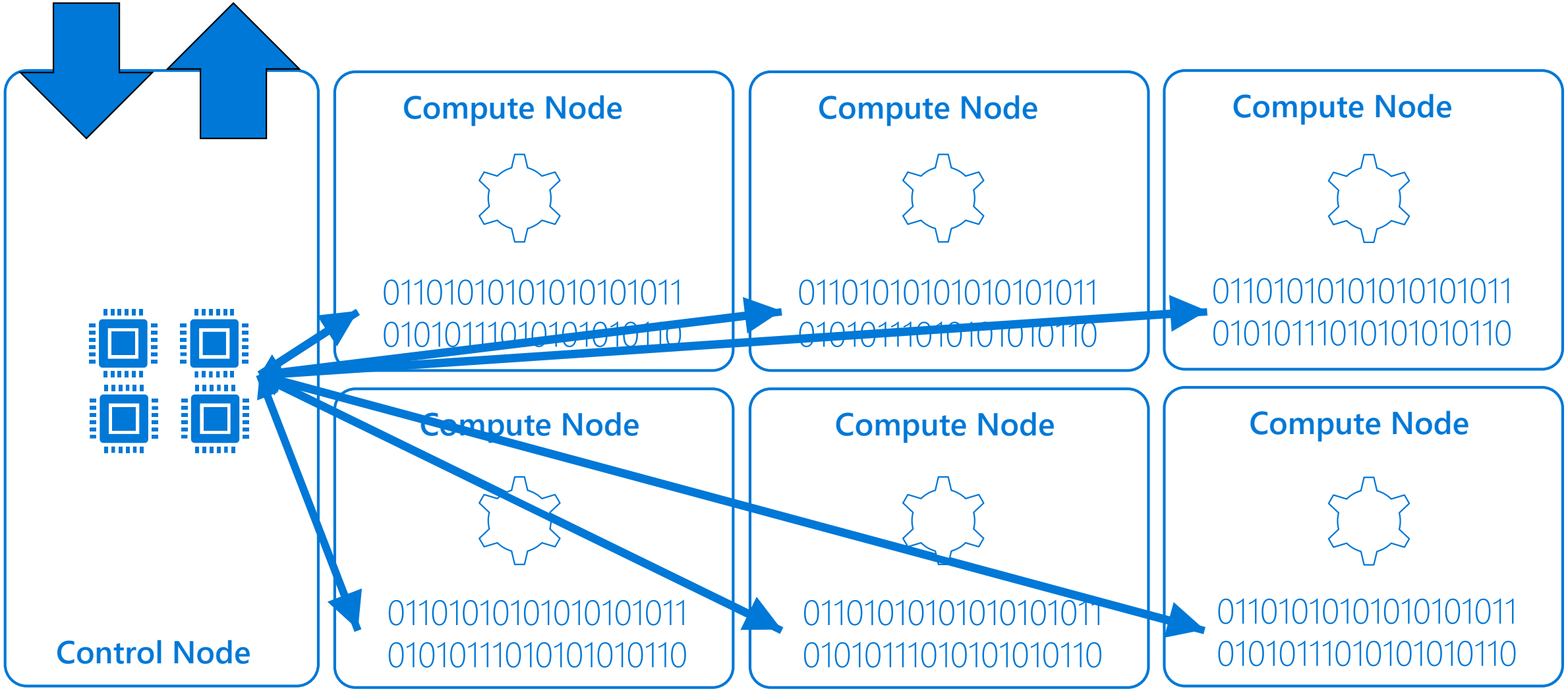
Advanced analytics

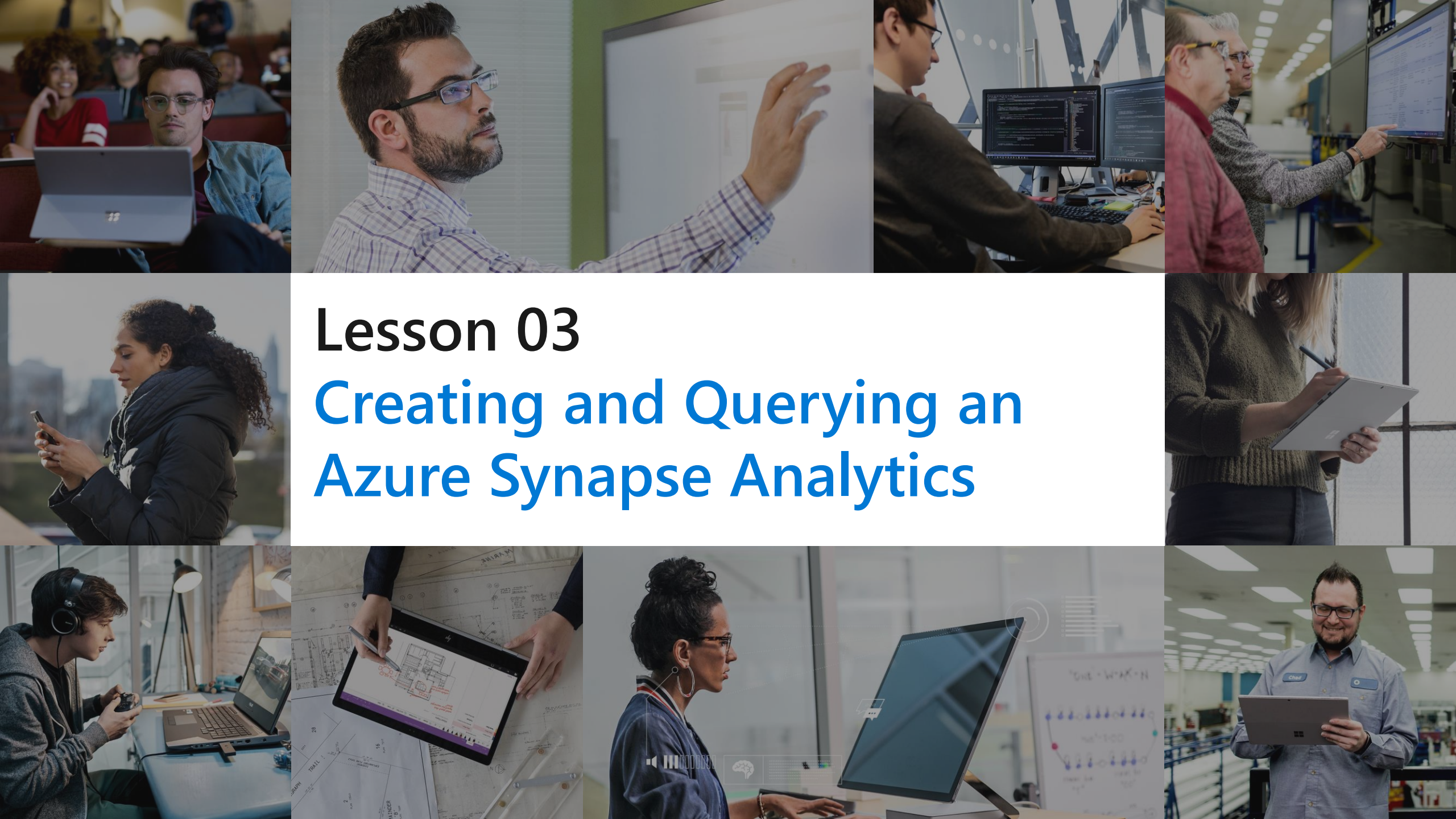
"We're trying to predict when our customers churn"

Real-time analytics

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

Massively Parallel Processing (MPP) concepts





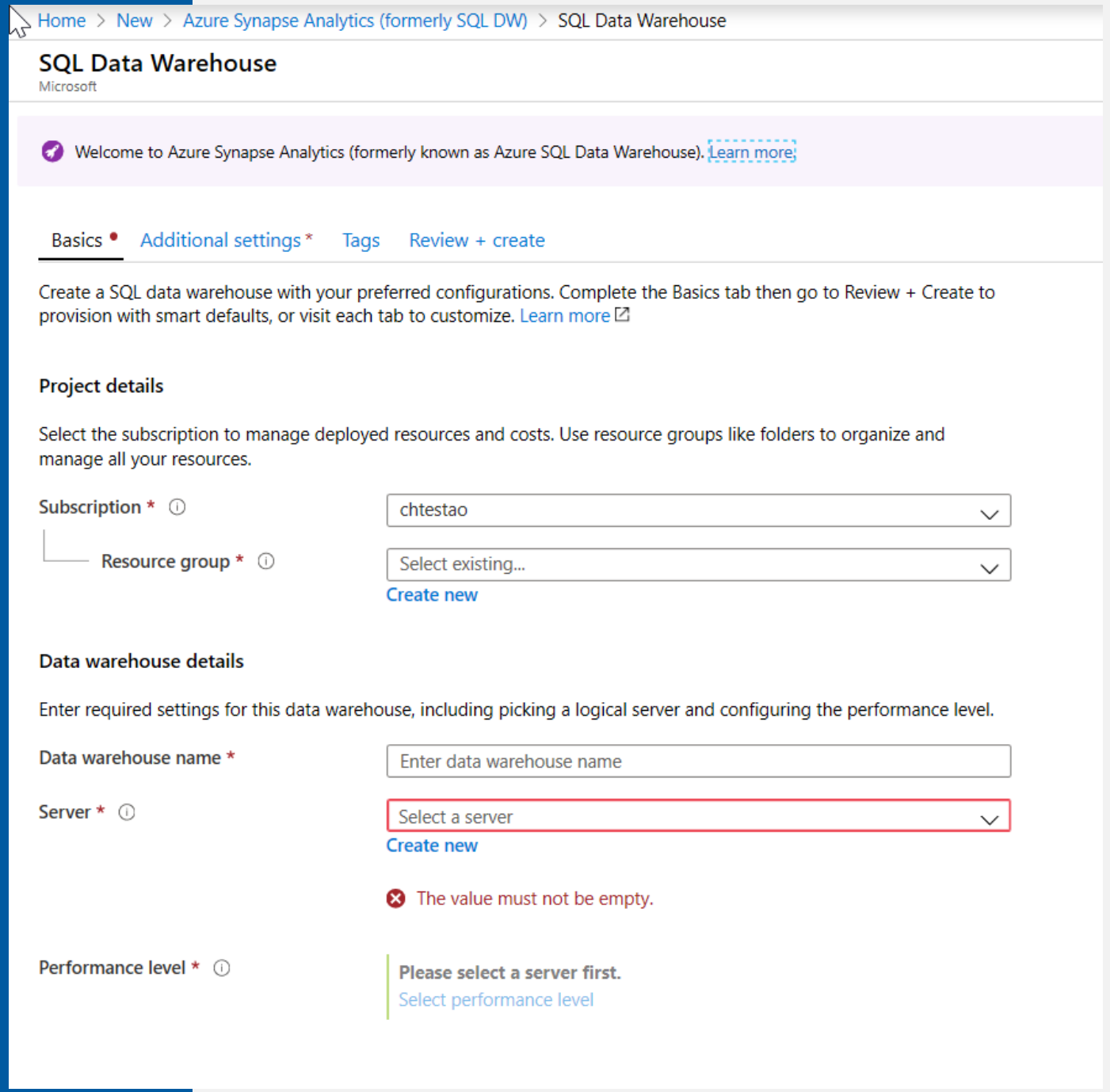
Lesson 03

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



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 )
                  REPLICATE | ROUND_ROBIN }
            [ , <CTAS_table_option> [ ,...n ] ]
    )
    AS <select_statement> [;]
```

Example

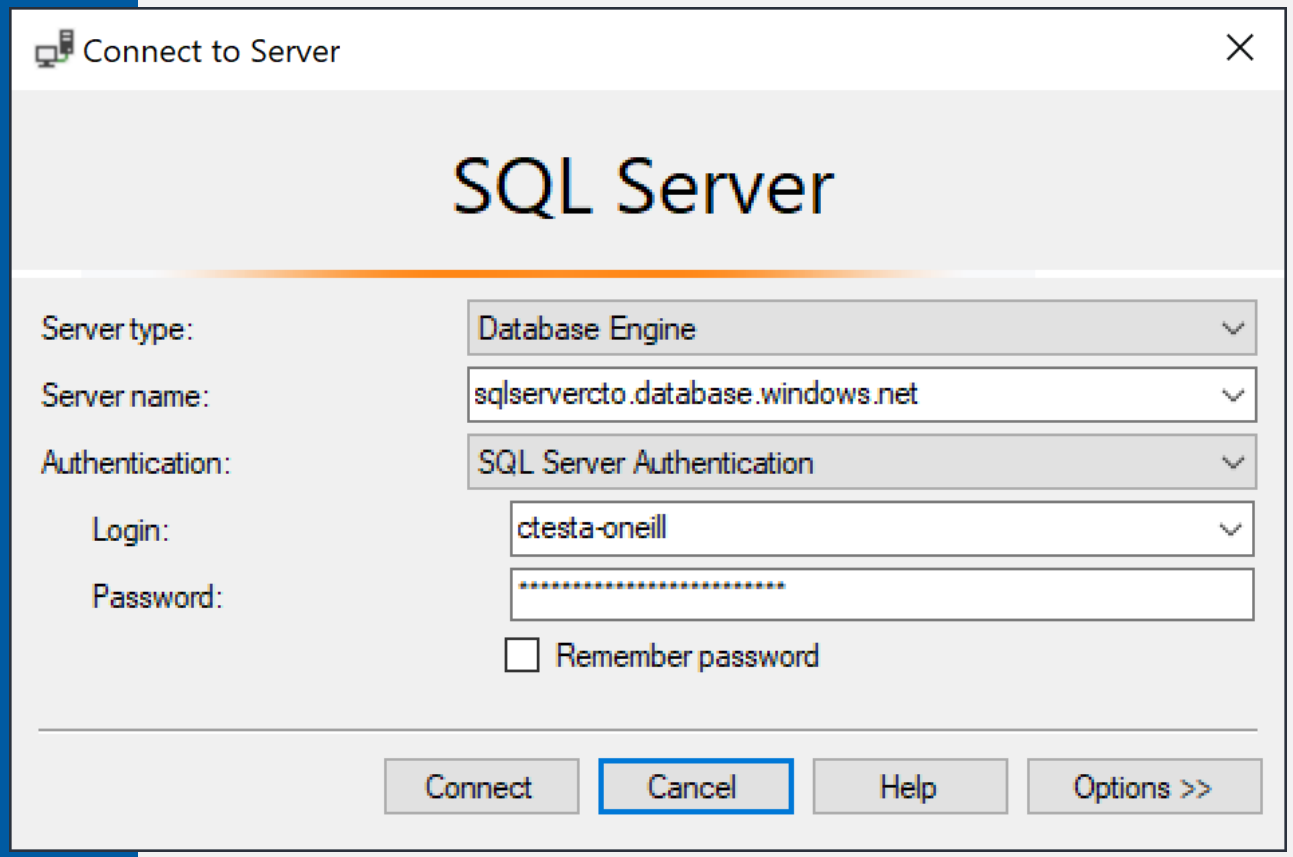
CREATE TABLE FactInternetSales_Copy

WITH

(DISTRIBUTION = HASH(SalesOrderNumber))

AS SELECT * FROM FactInternetSales

Querying with different client applications.



Lesson 04

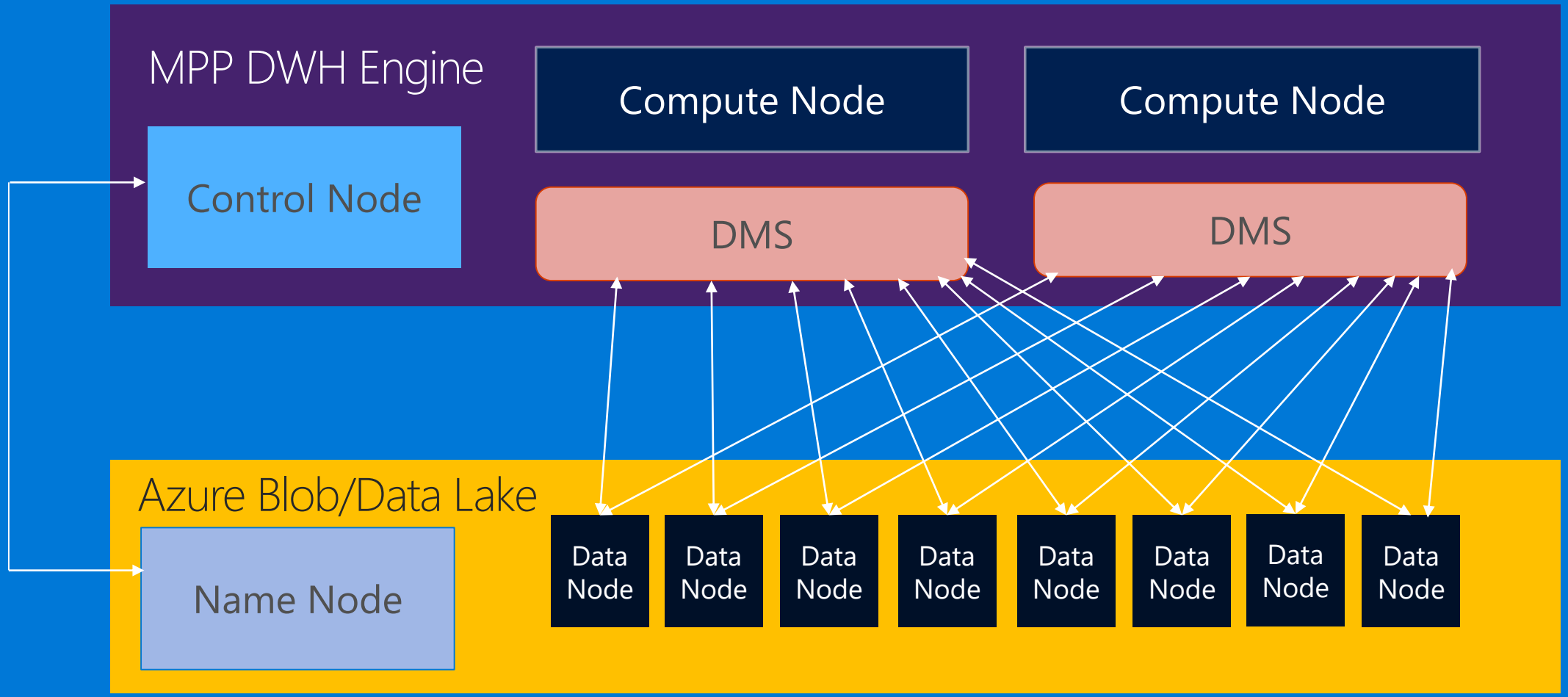
Using PolyBase to Load Data in Azure Synapse Analytics

Lesson Objectives

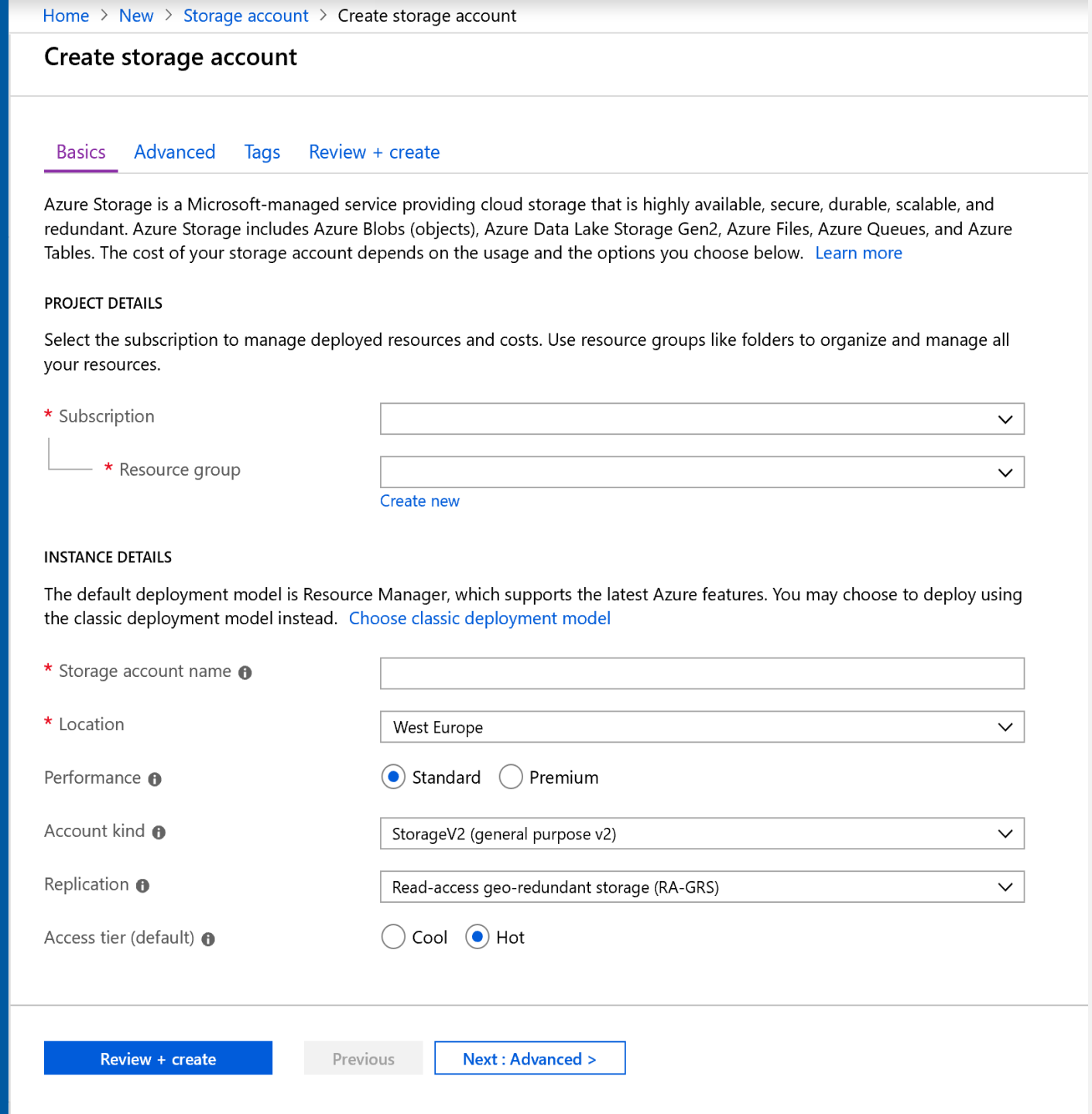
- Explore how PolyBase works
- Upload text data to Azure Blob store
- Collect the security keys for Azure Blob store
- Create an Azure Synapse Analytics
- Import data from Blob Storage to the Data Warehouse

How PolyBase works

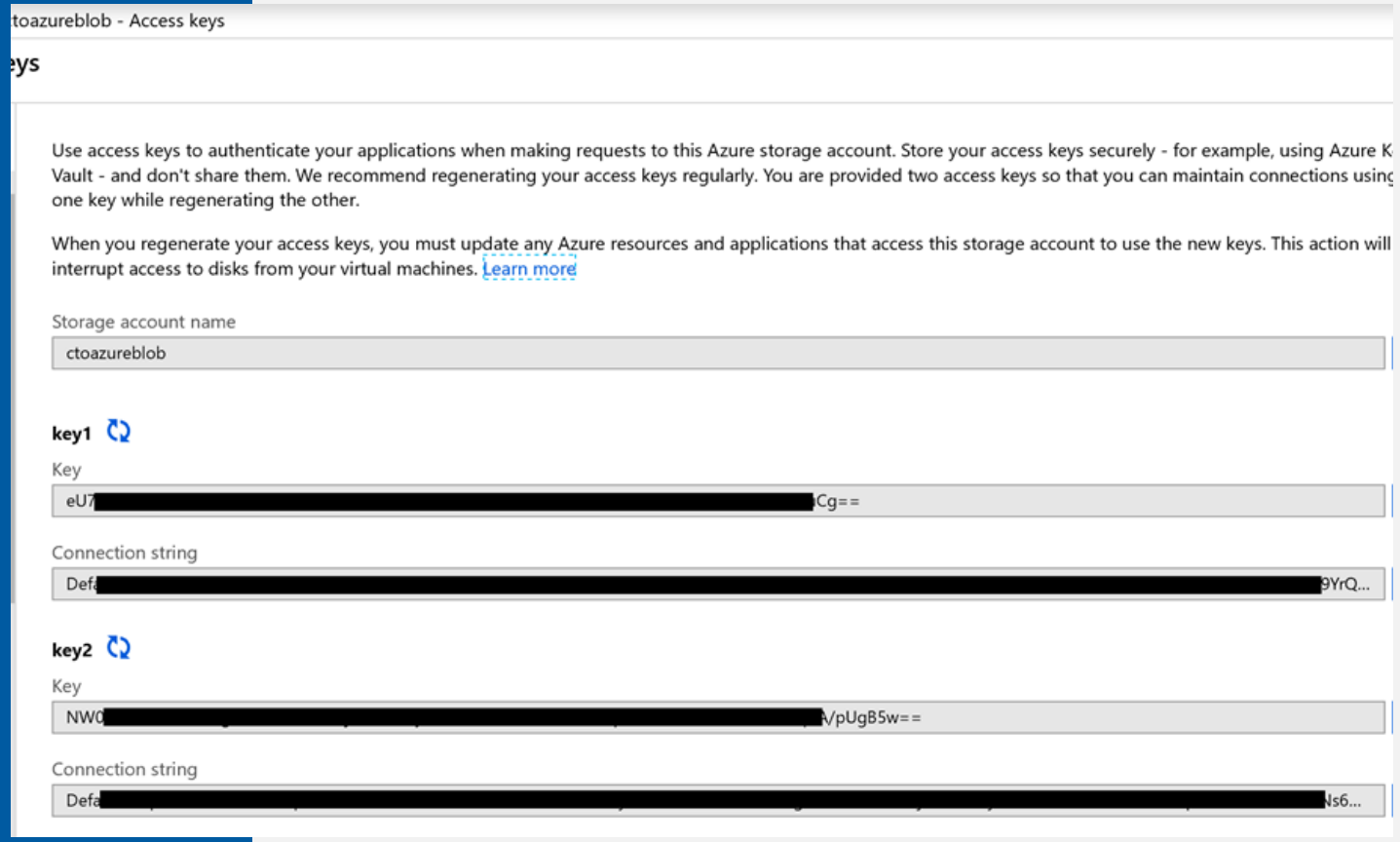
The MPP engine's integration method with PolyBase



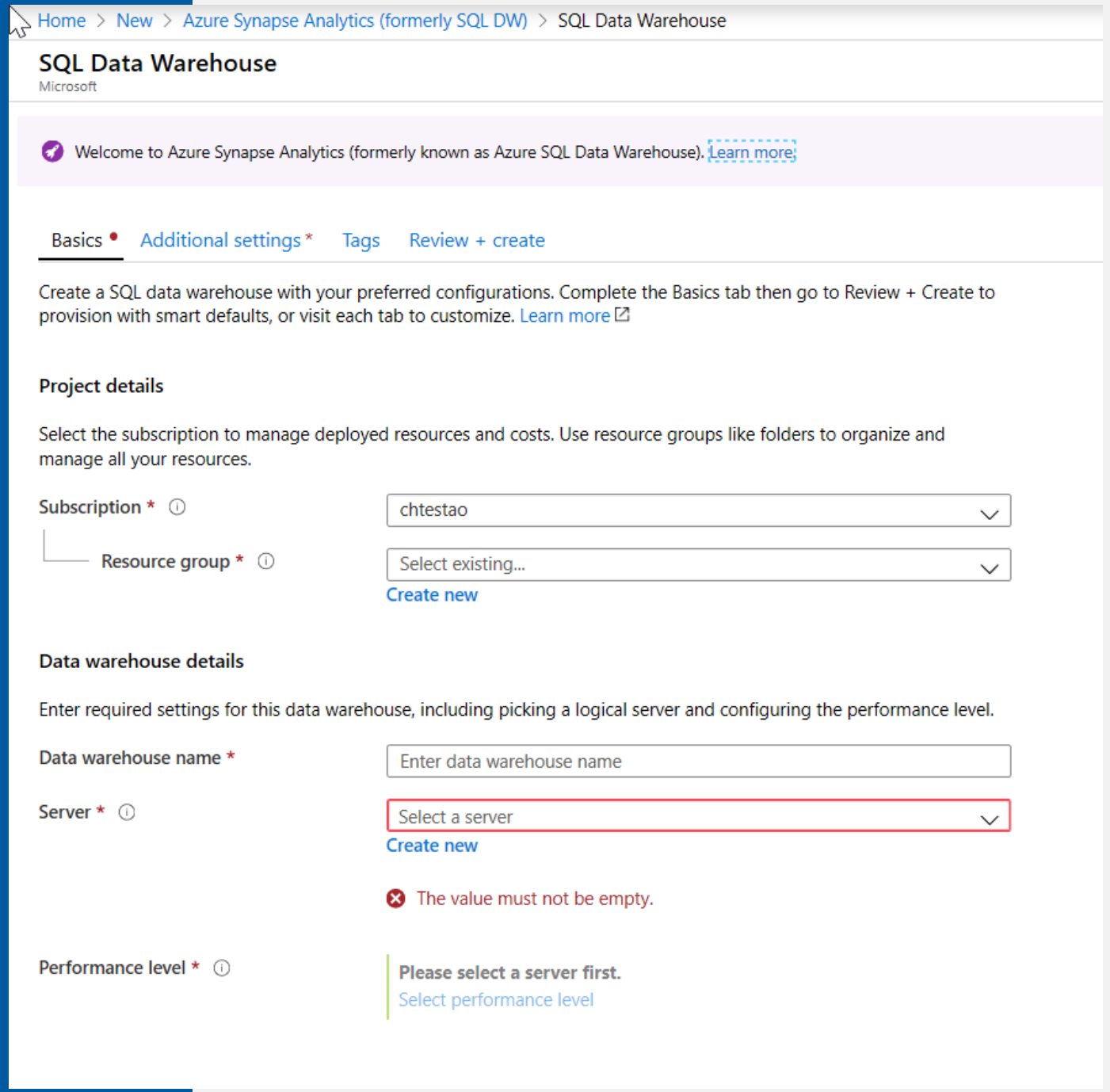
Upload text data to Azure Blob store



Collect the Storage keys



Create an Azure Synapse Analytics



Lab: Working with Relational Data Stores in the Cloud

Microsoft

Module 06: Performing Real-Time Analytics with Stream Analytics

Agenda

- L01 - Data streams and event processing
- L02 - Data ingestion with Event Hubs
- L03 - 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 • Systems • Applications	• Event Hubs • IoT Hubs • Azure Blob Store	• Stream Analytics Query Language • .NET SDK	• Azure Data Lake • Cosmos DB • SQL Database • Blob Store • Power BI

Lesson 02

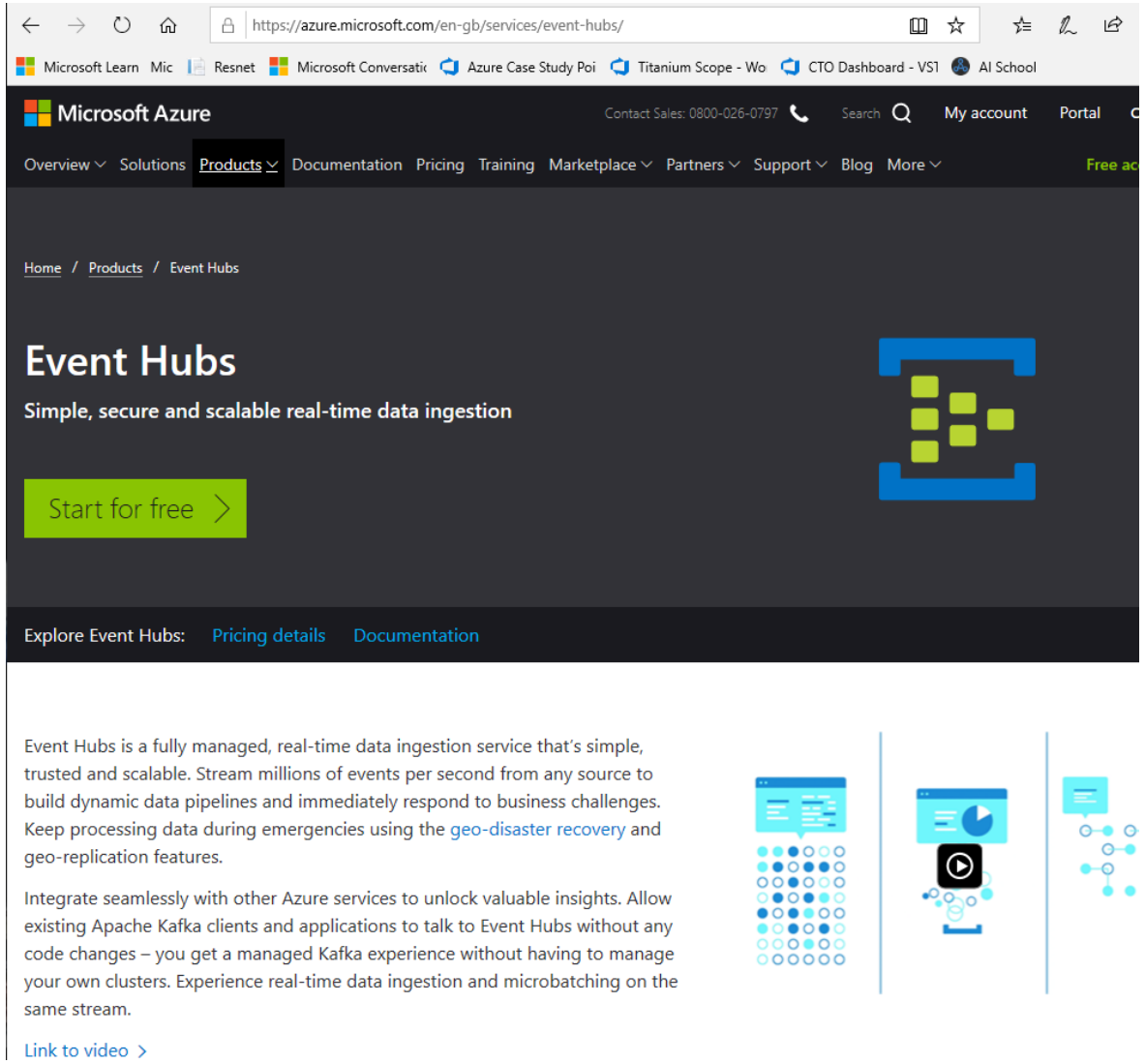
Data Ingestion with Event Hubs

Lesson Objectives

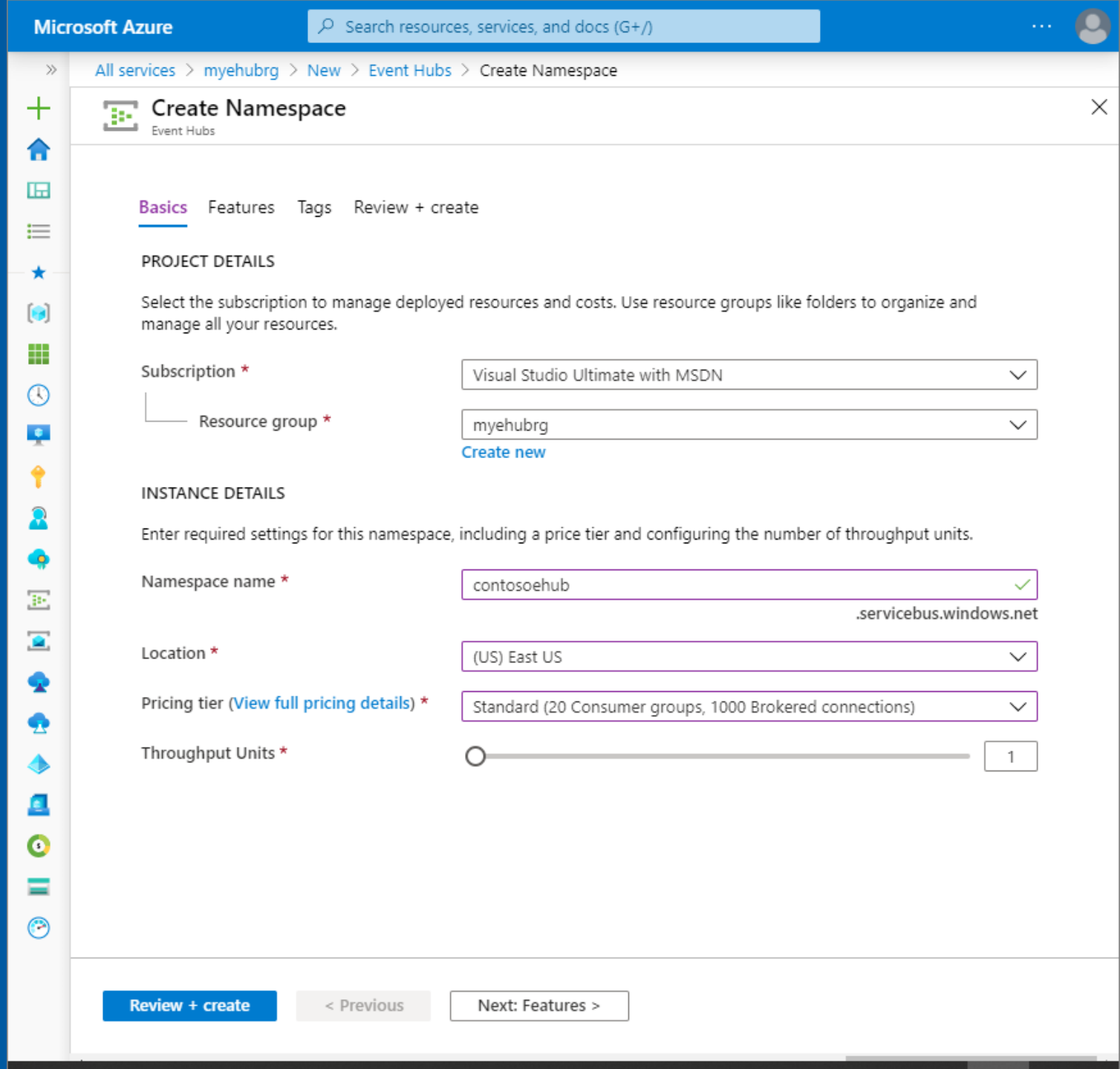
- Describe Azure Event Hubs
- Create an Event Hub
- Evaluate the performance of an Event Hub
- Configure applications to use an Event Hub

Azure Event Hubs

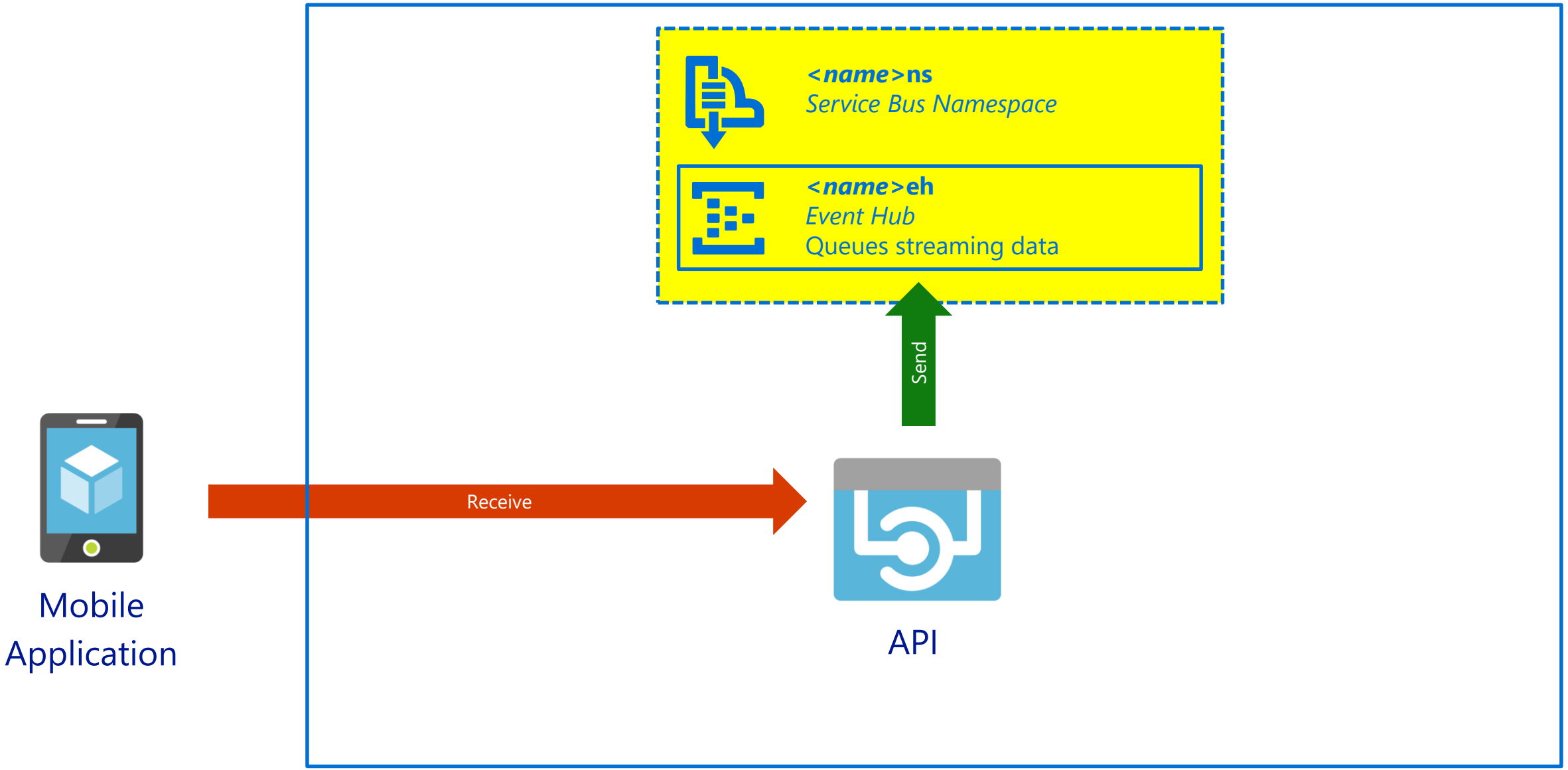
"Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple applications."



Create an Event Hub



Configure Applications to use Event Hubs



Lesson 03

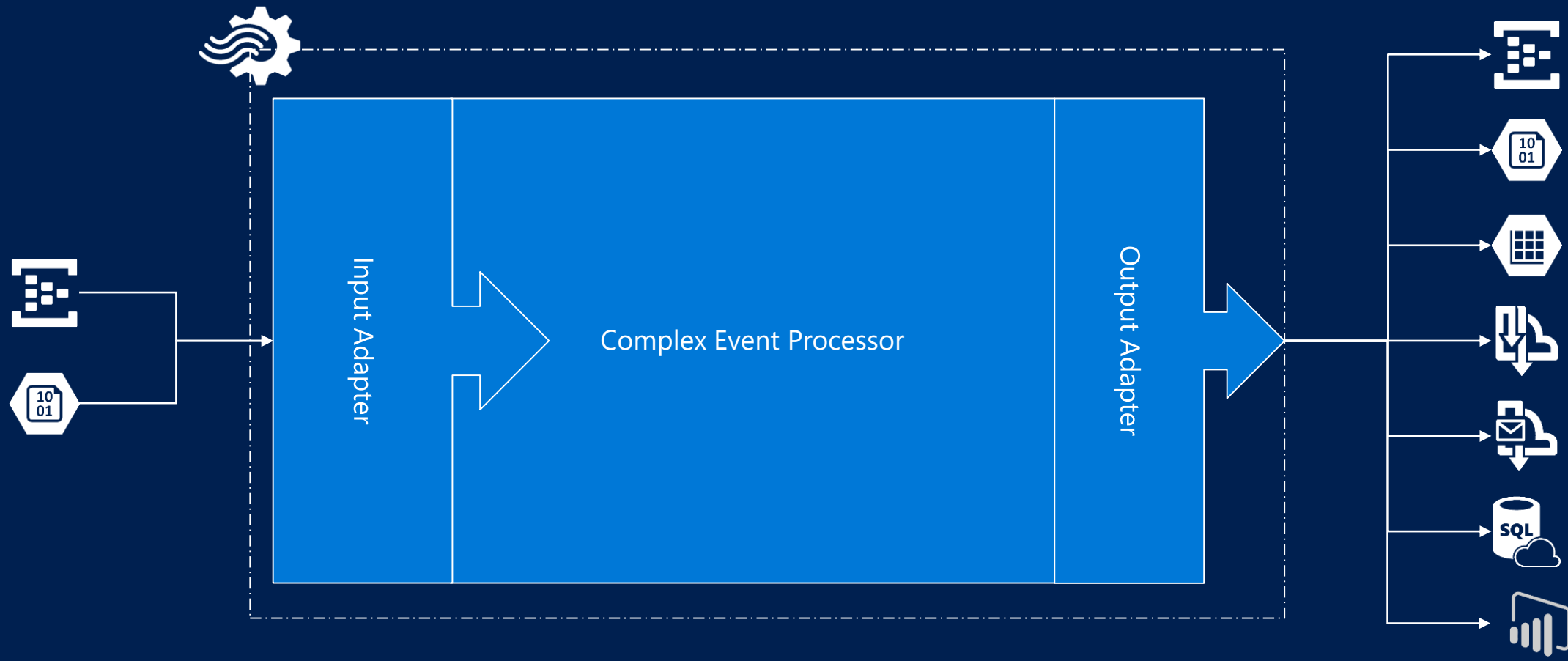
Processing Data with Stream Analytics Jobs

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

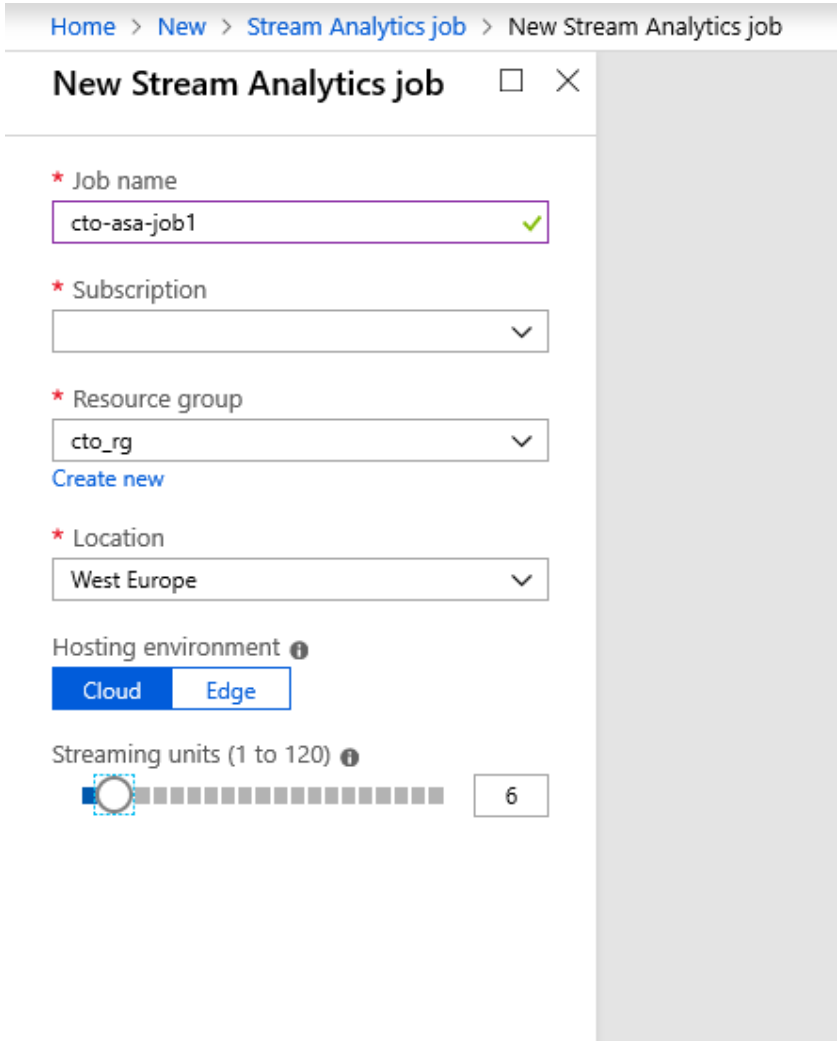
Azure Stream Analytics Workflow

Complex Event Processing of Stream Data in Azure

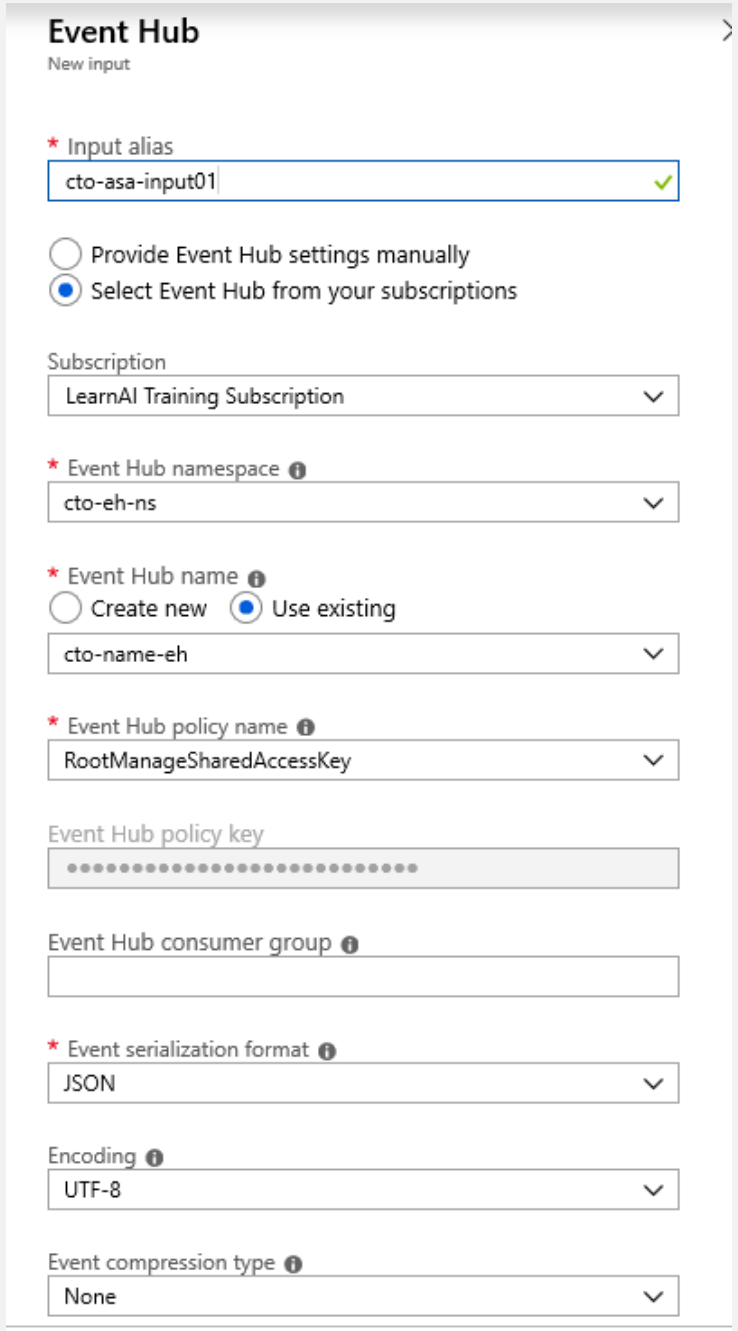


Create Stream Analytics Service

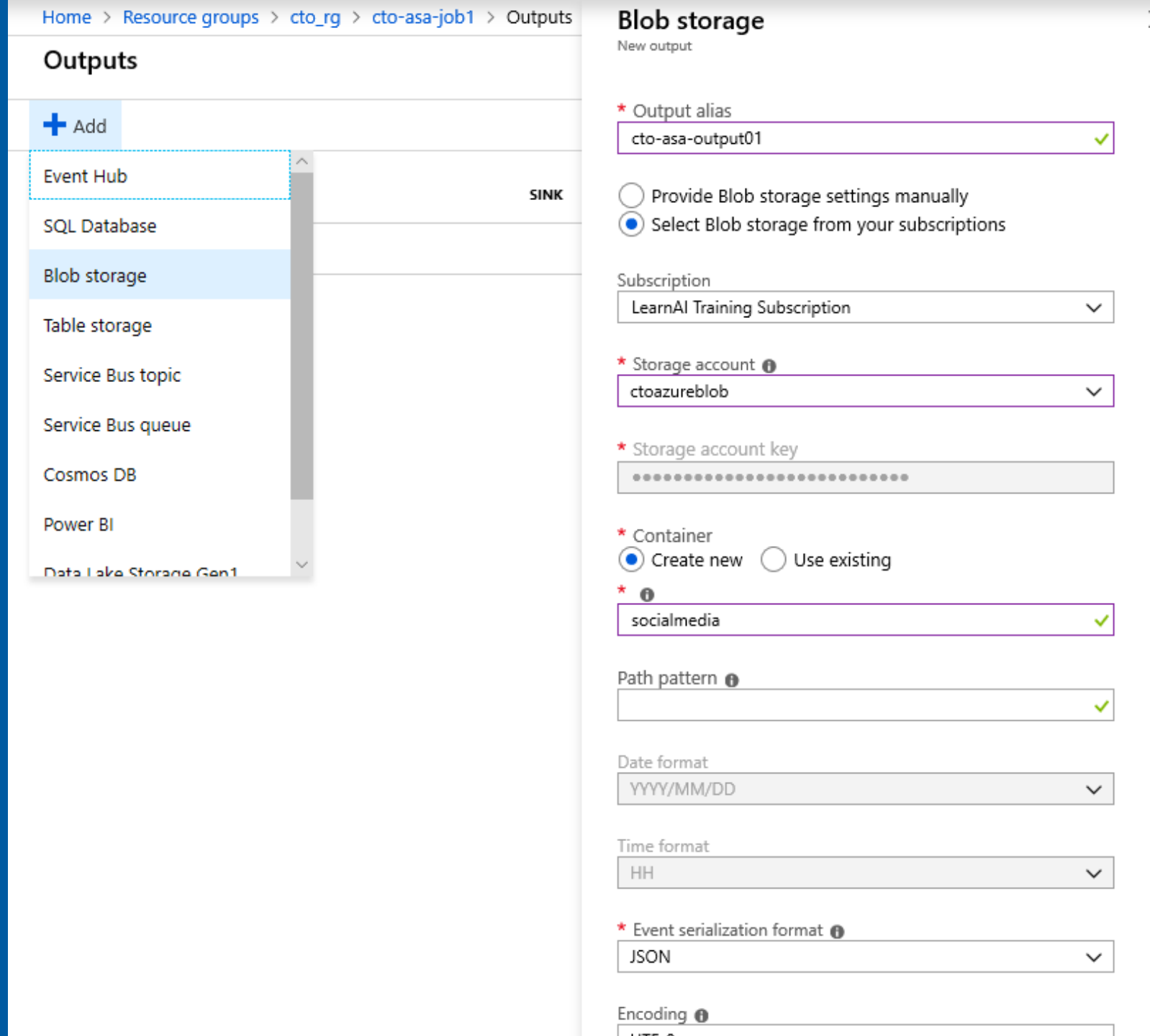
- Job name
- Subscription
- Resource group
- Location



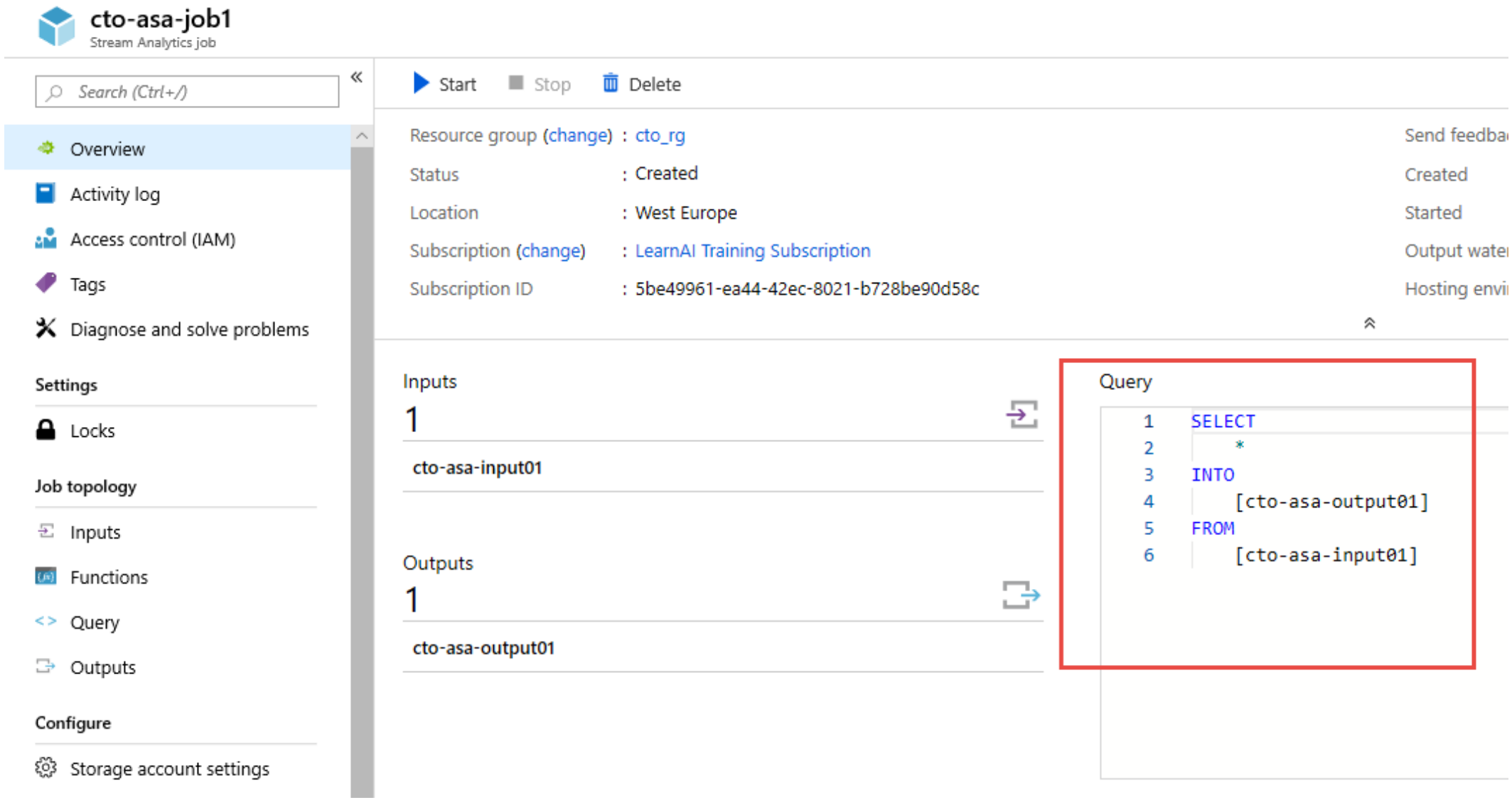
Create a Stream Analytics Job Input.



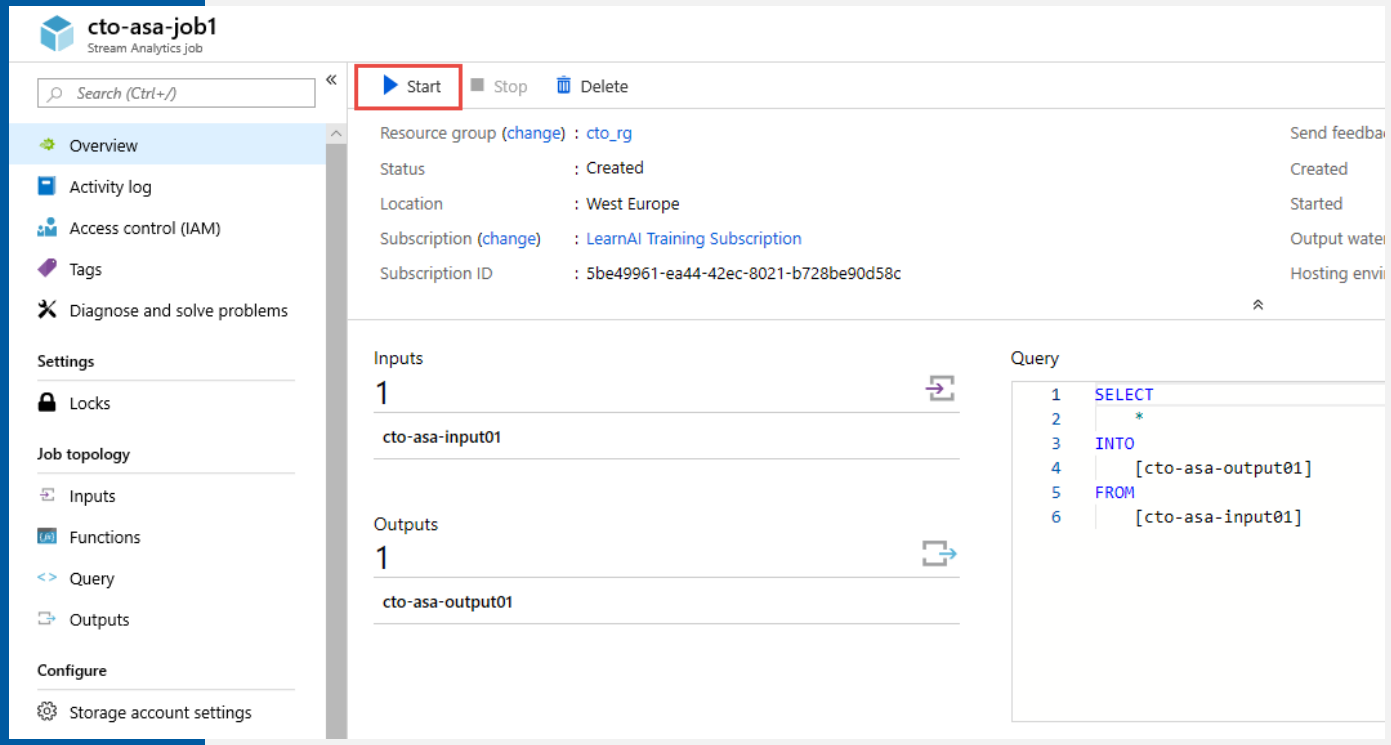
Create a Stream Analytics Job Output.



Write a transformation query



Start a Stream Analytics Job



Lab: Performing Real-Time Analytics with Stream Analytics

Microsoft

Module 07: Orchestrating Data Movement with Azure Data Factory

Agenda

- L01 - Introduction to Azure Data Factory
- L02 - Understand Azure Data Factory components
- L03 - Integrate Azure Data Factory with Databricks

Lesson 01

Introduction to Azure Data Factory

Lesson Objectives

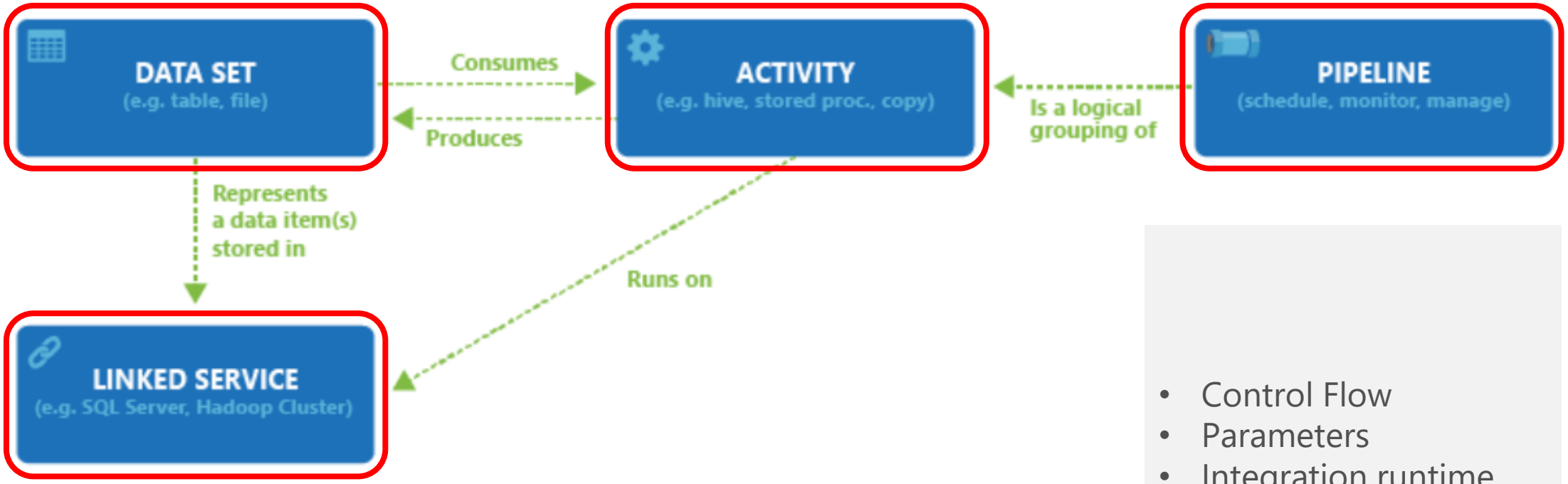
- What is Azure Data Factory
- The Data Factory process
- Azure Data Factory components
- Azure Data Factory security

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.

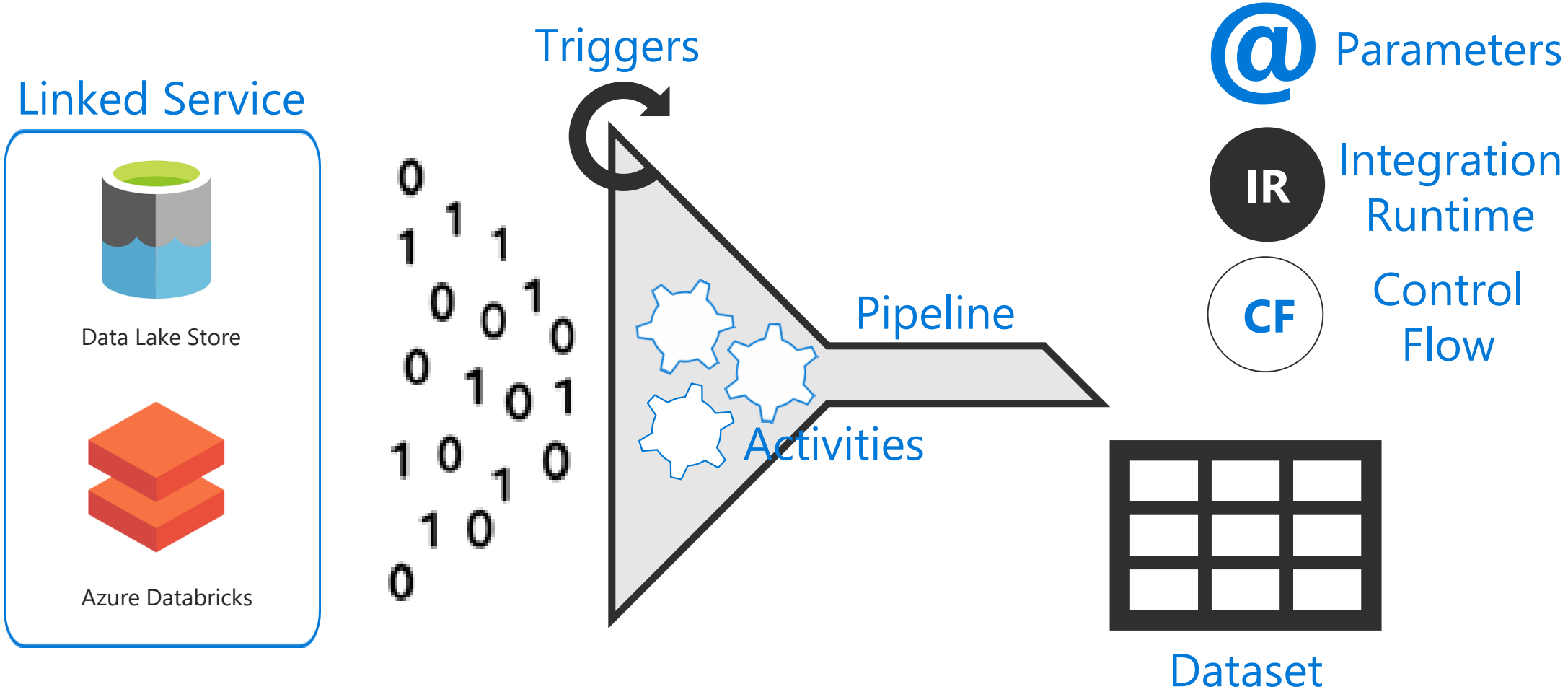
Lesson 02

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**.

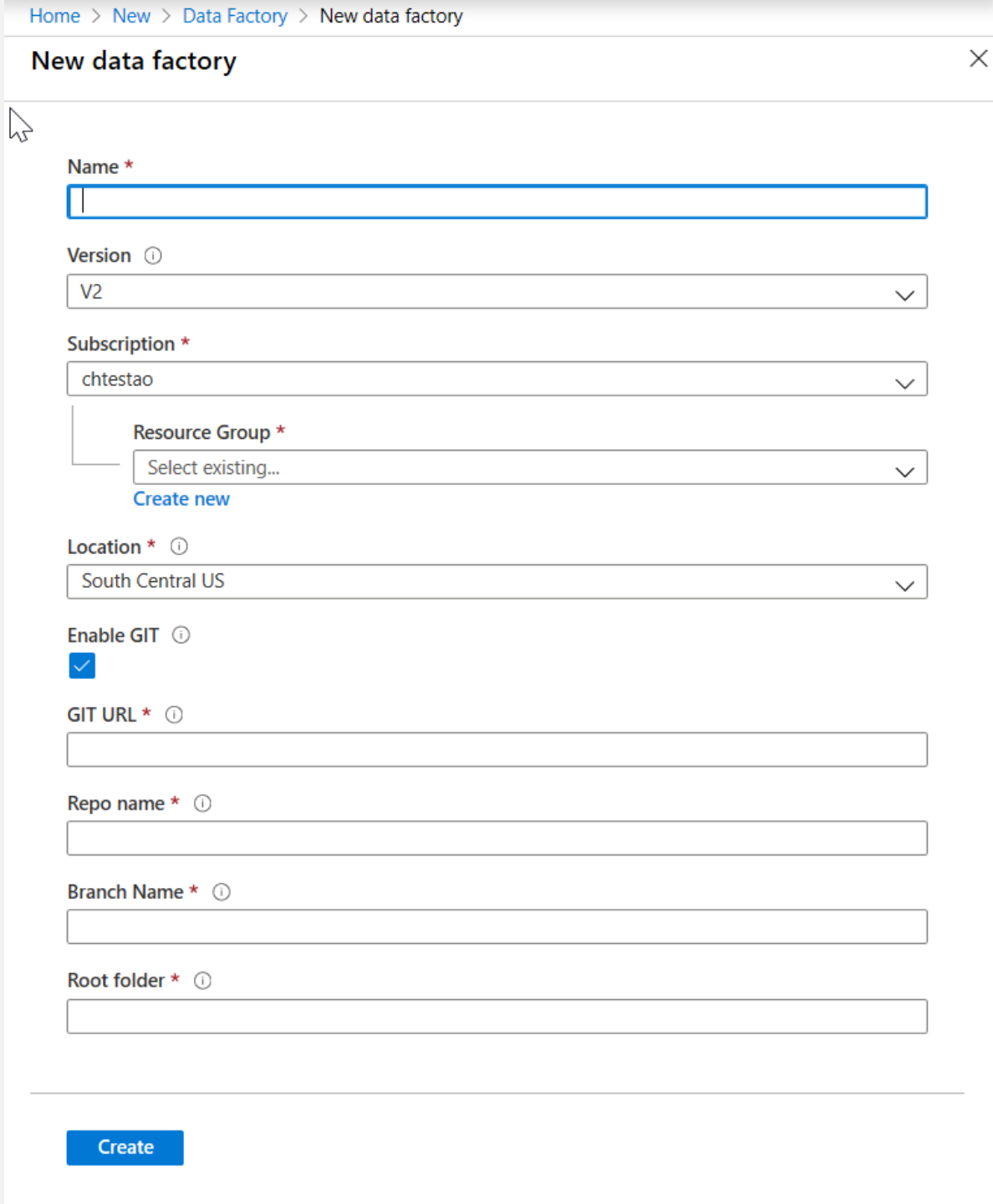
Lesson 03

Ingesting and Transforming data

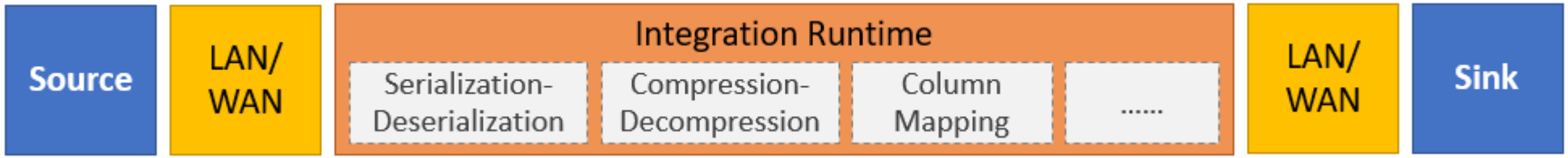
Lesson Objectives

- How to setup Azure Data Factory
- Ingest data using the Copy Activity
- Transforming data with the Mapping Data Flow

Create Azure Data Factory



Ingesting data with the Copy Activity

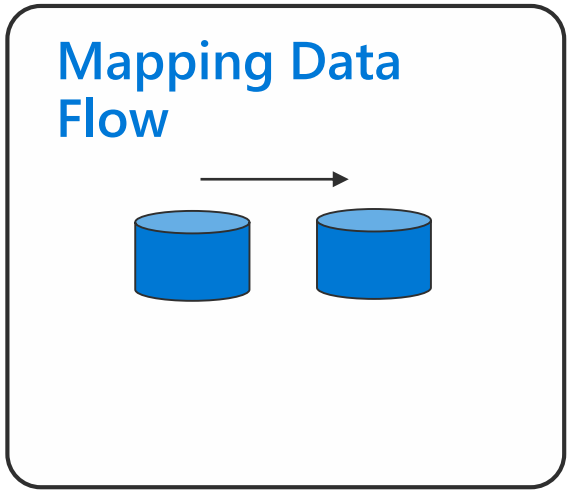


- Reads data from a source data store.
- Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration of the input dataset, output dataset, and Copy activity.
- Writes data to the sink/destination data store

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



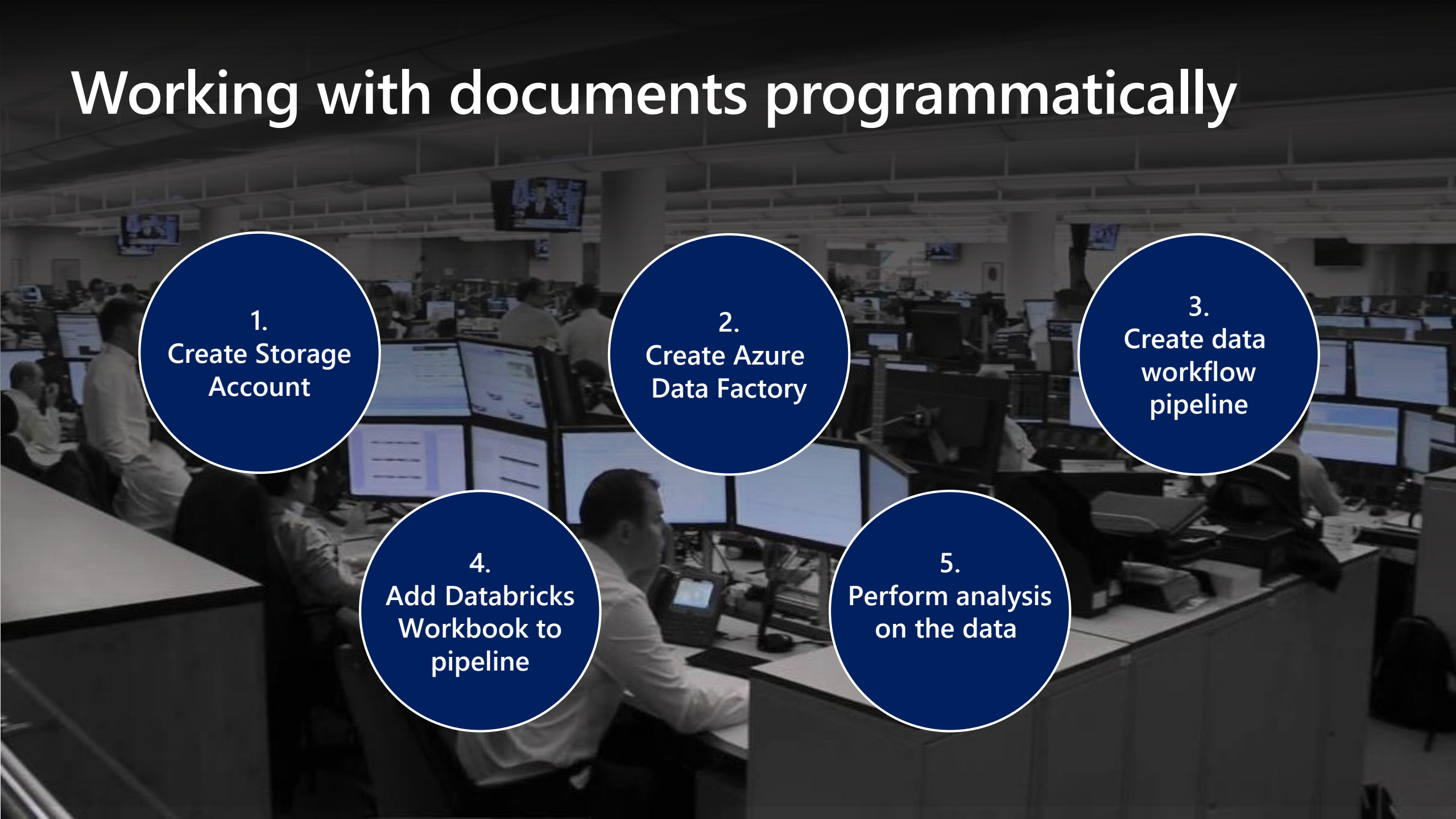
Lesson 04

Integrate Azure Data Factory with Azure Databricks

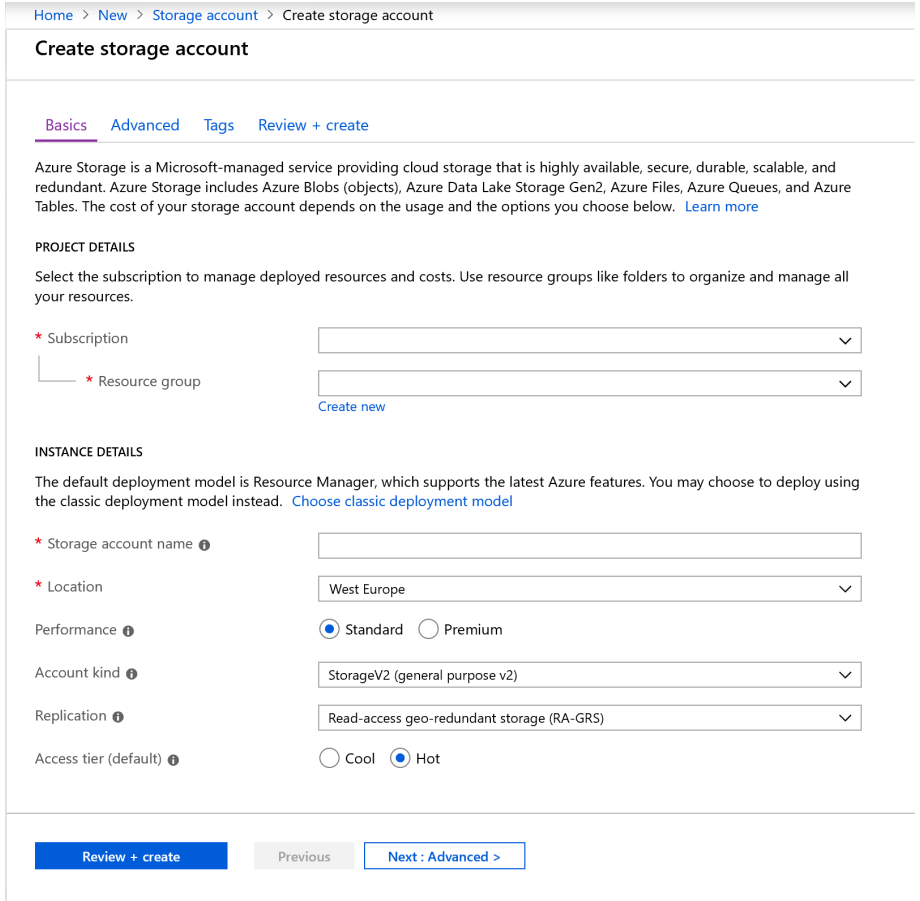
Lesson Objectives

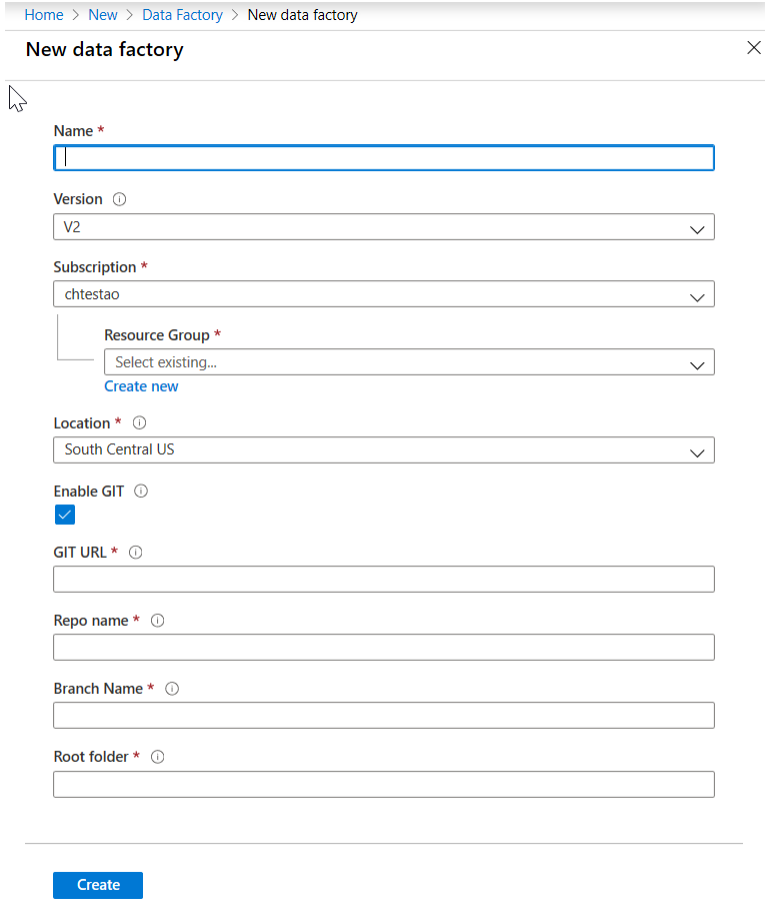
- Use Azure Data Factory (ADF) to ingest data and create an ADF pipeline.
- Create Azure Storage account and the Azure Data Factory instance
- Use ADF to orchestrate data transformations using a Databricks Notebook activity.

Working with documents programmatically

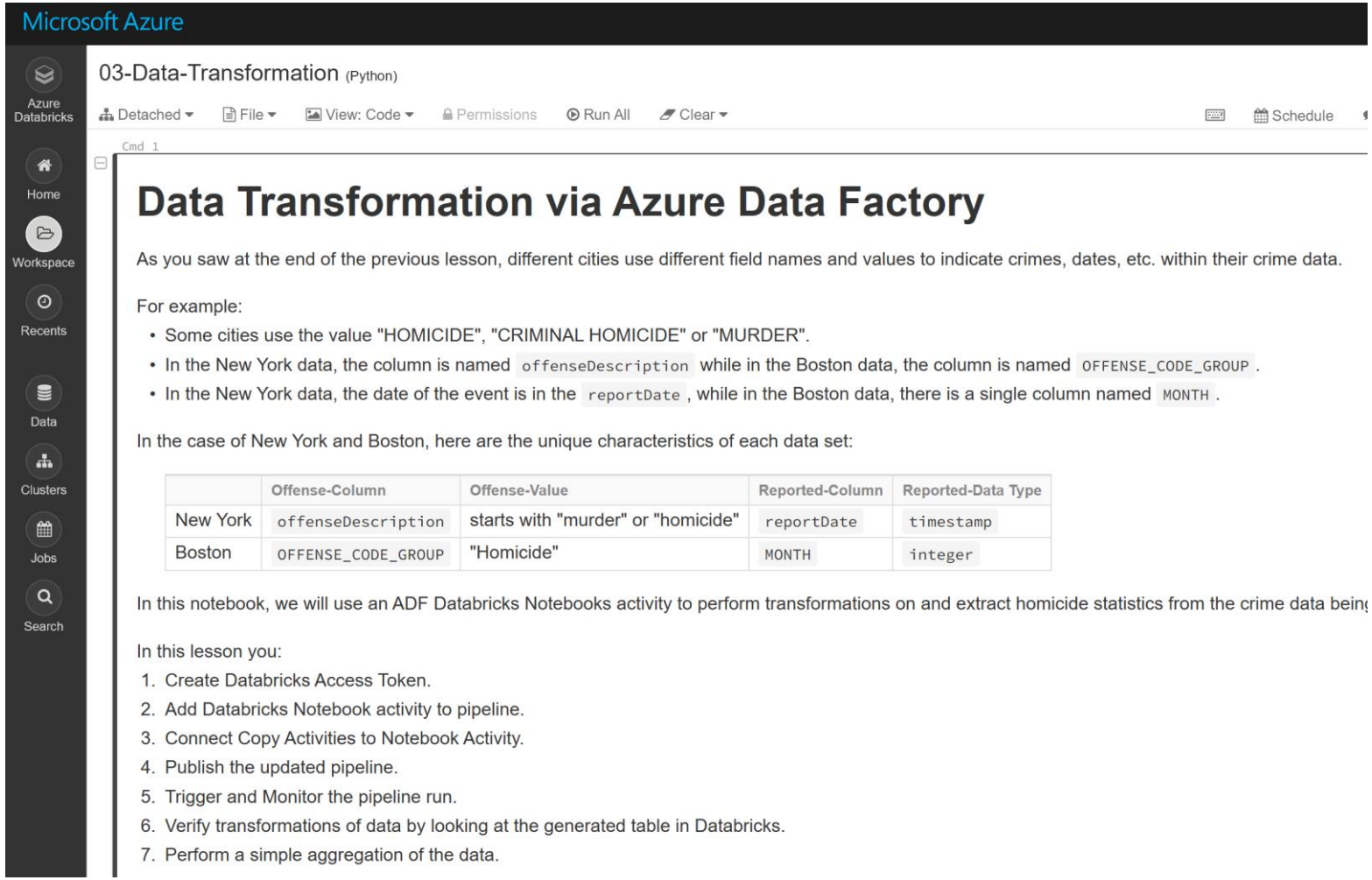


Create Azure Storage account and the Azure Data Factory Instance





Use ADF to orchestrate data transformations using a Databricks Notebook activity



Lab: Orchestrating Data Movement with Azure Data Factory

Microsoft

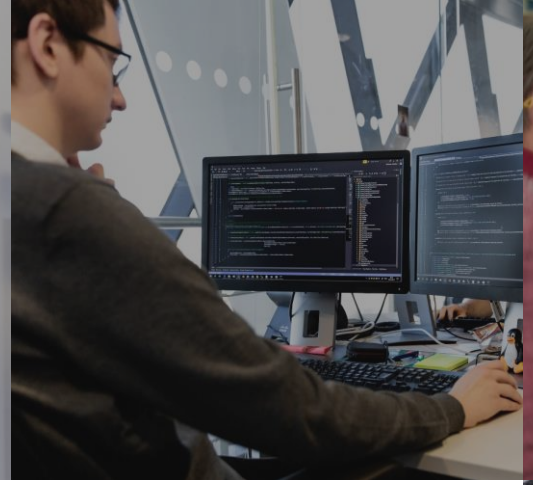
Module 08: Securing Azure Data Platforms

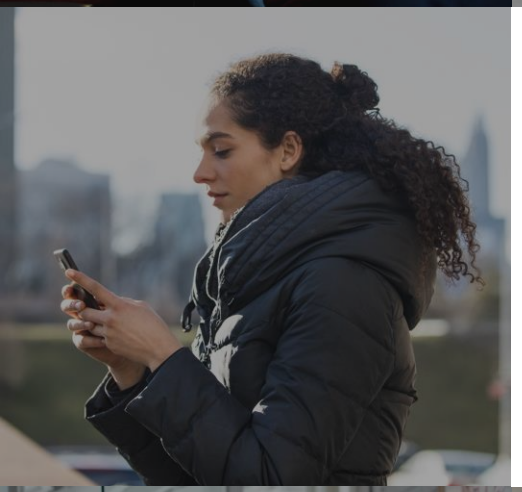
13.30 น.

Agenda

- L01 - An introduction to security
- L02 - Key security components
- L03- Securing Storage Accounts and Data Lake Storage
- L04 - Securing data stores



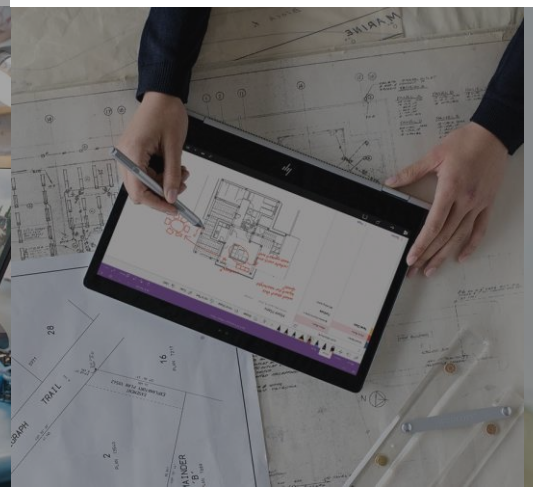
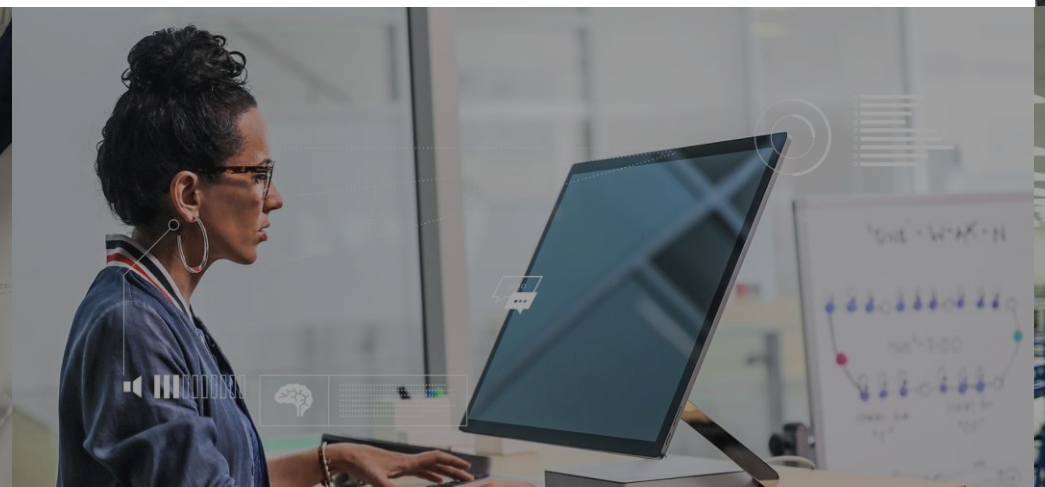




Lesson 01

An Introduction to Security

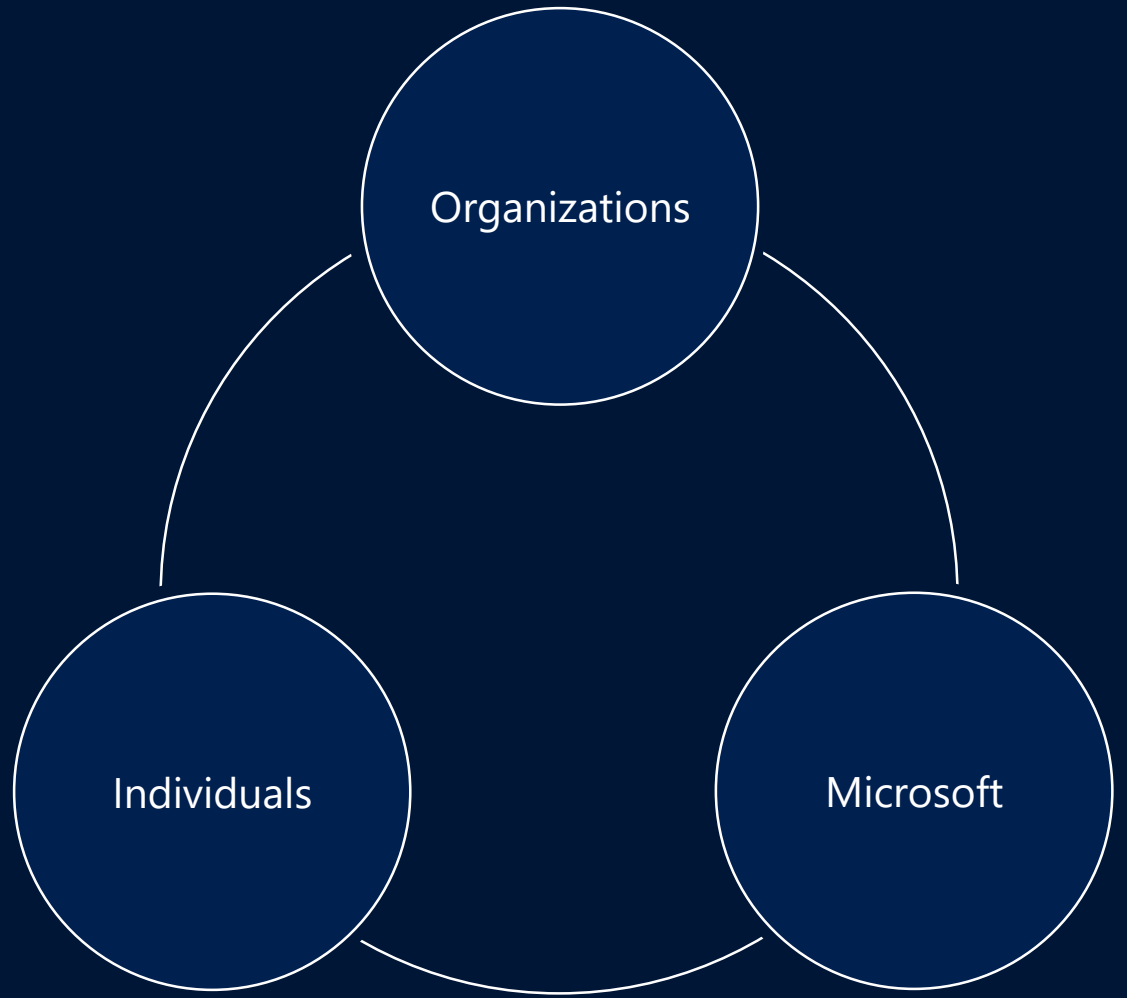


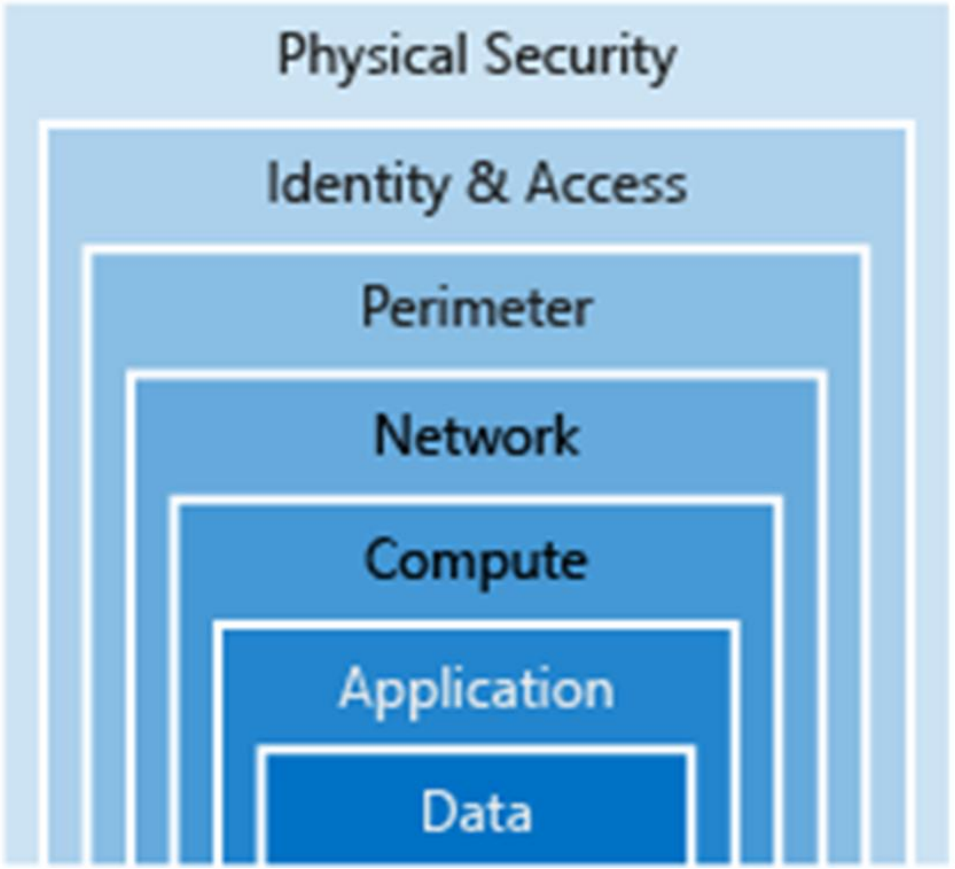

Lesson Objectives

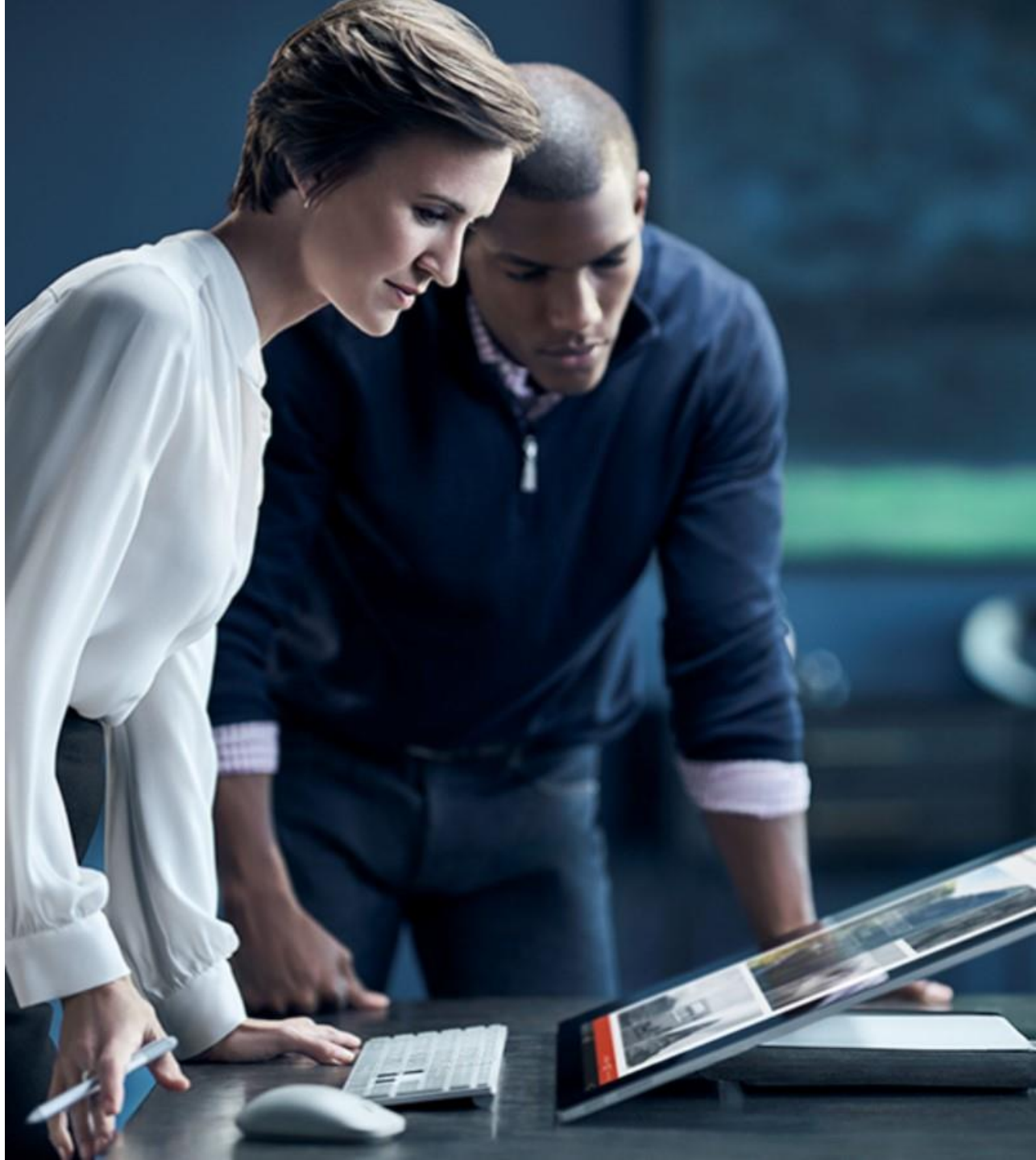
- Shared Security Responsibility
- A layered approach to security
- The Azure Security Center
- Azure Government

Shared Security Responsibility



A Layered Approach to Security.





Azure Security Center

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

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

Identity and Access

Authentication

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

Authorization

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

.

Azure Active Directory Features.

Single Sign-On

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

Apps & Device Management

You can manage your cloud and on-premises apps and devices and the access to your organizations 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

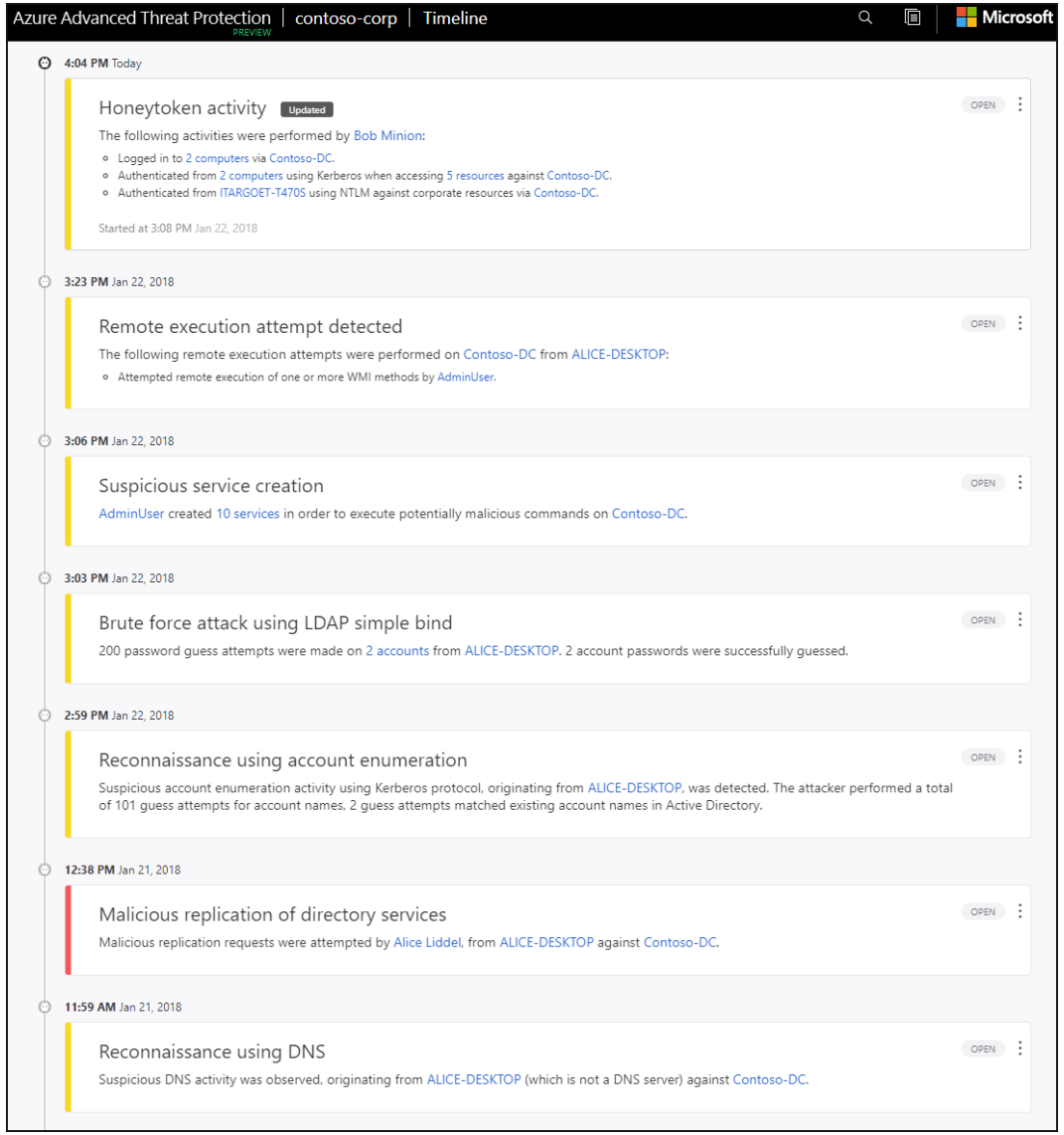
Enables the encryption of databases using:

- Transparent Data Encryption

Encrypting Secrets

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

Azure Threat Protection



Lesson 03

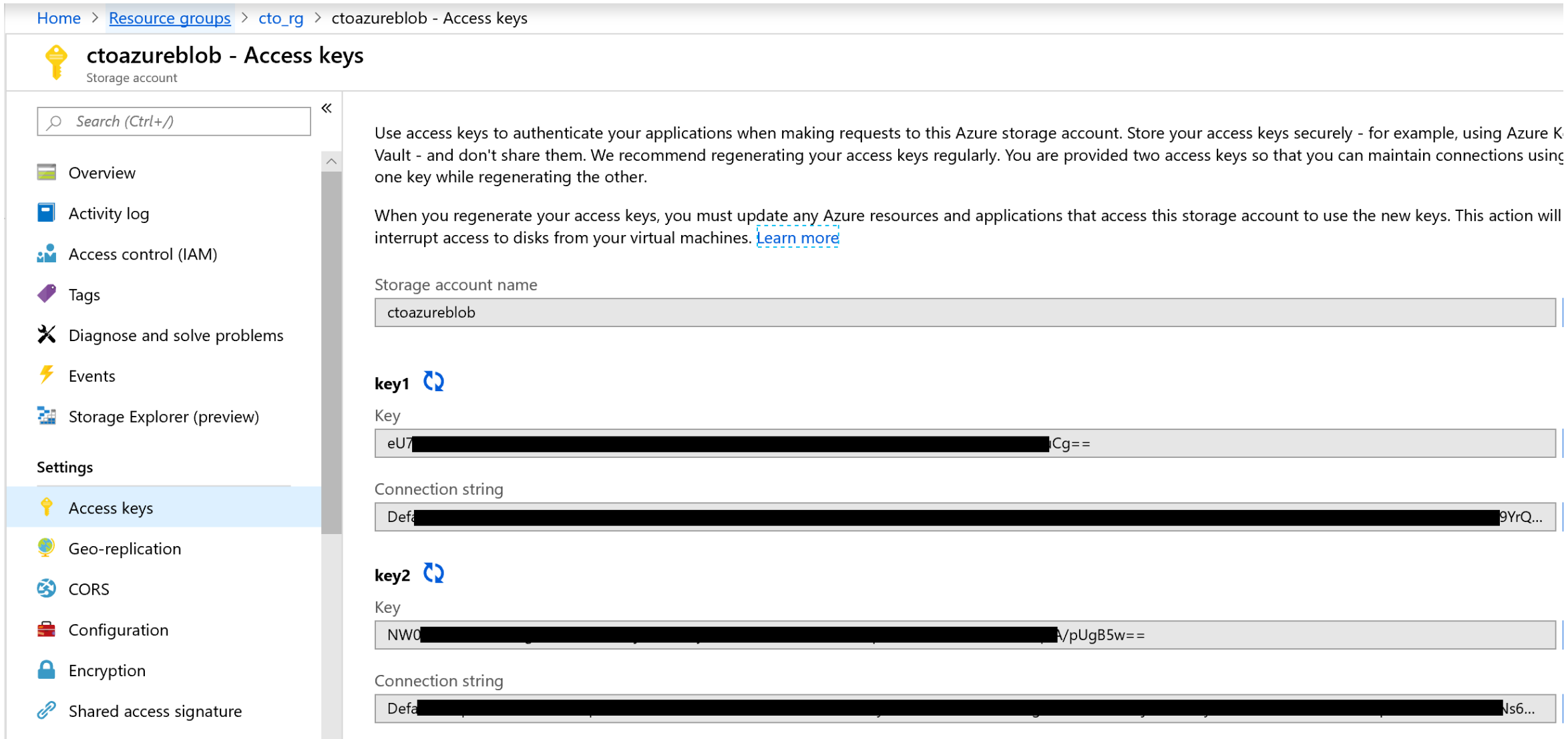
Securing Storage Accounts and Data Lake Storage

Lesson Objectives

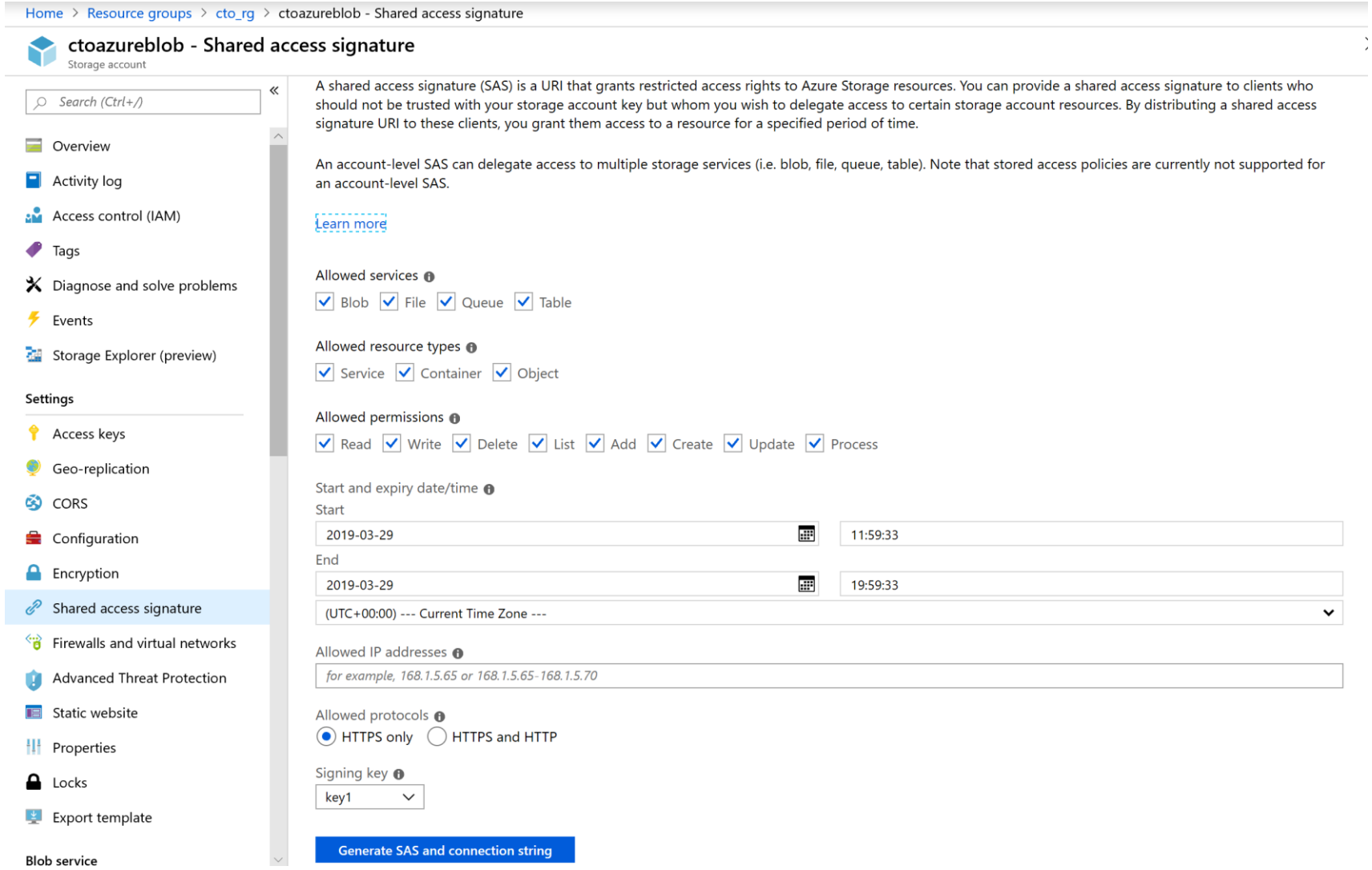
- Storage Account security features
- Explore the authentication options available to access data
  - Storage Account Key
  - Shared Access Signature
- Control network access to the data
- Managing encryption
- Azure Data Lake Storage Gen II security features

Storage Account Security Features

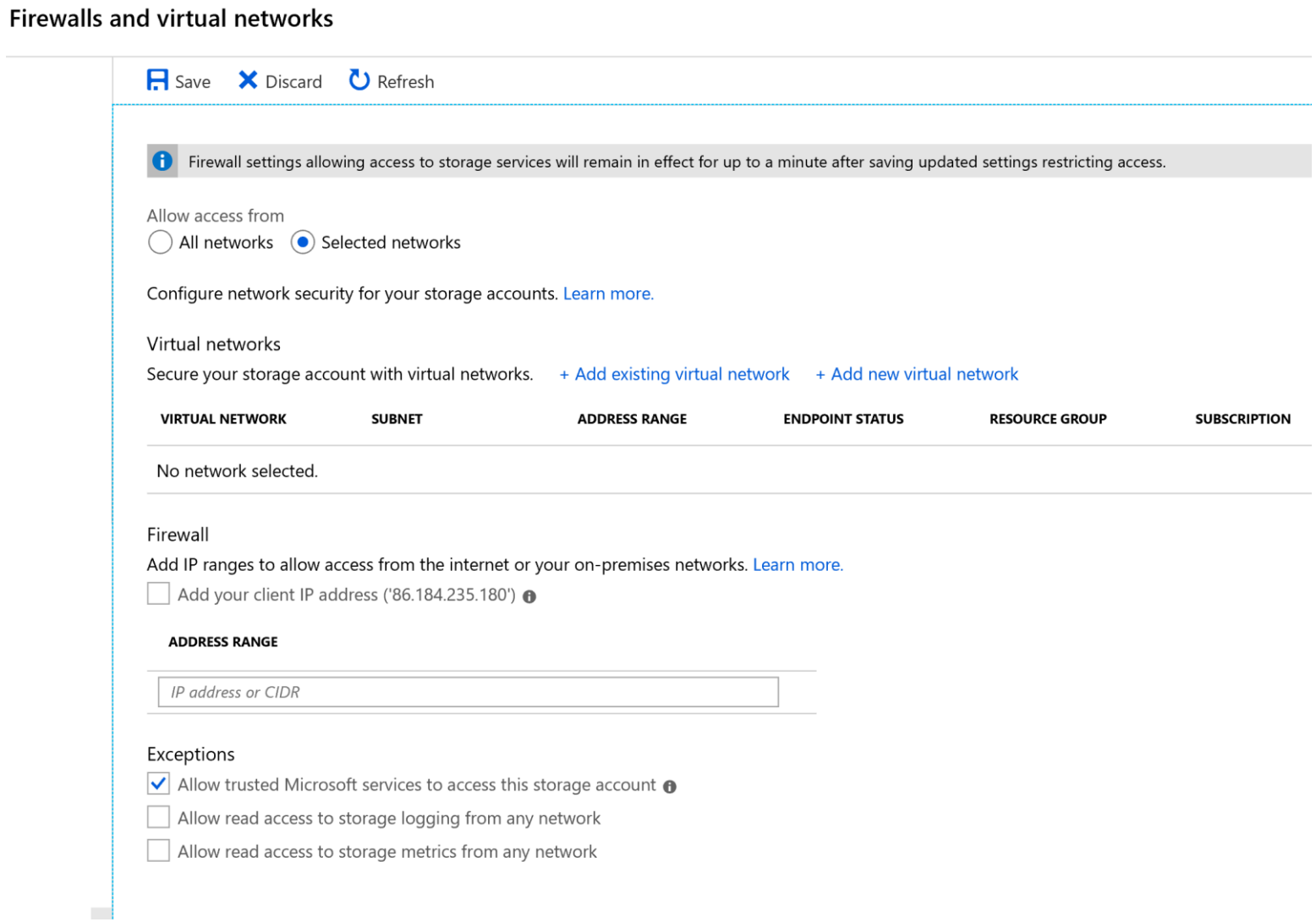
Storage Account Keys



Shared Access Signatures



Control network access to data



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

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

Auditing and Monitoring

Lab: Securing Azure Data Platforms

Microsoft

Module 09: Monitoring and Troubleshooting Data Storage and Processing

Agenda

L01 - General Azure monitoring capabilities

L02 - Troubleshoot common data storage issues

L03 - Troubleshoot common data processing issues

L04 - 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

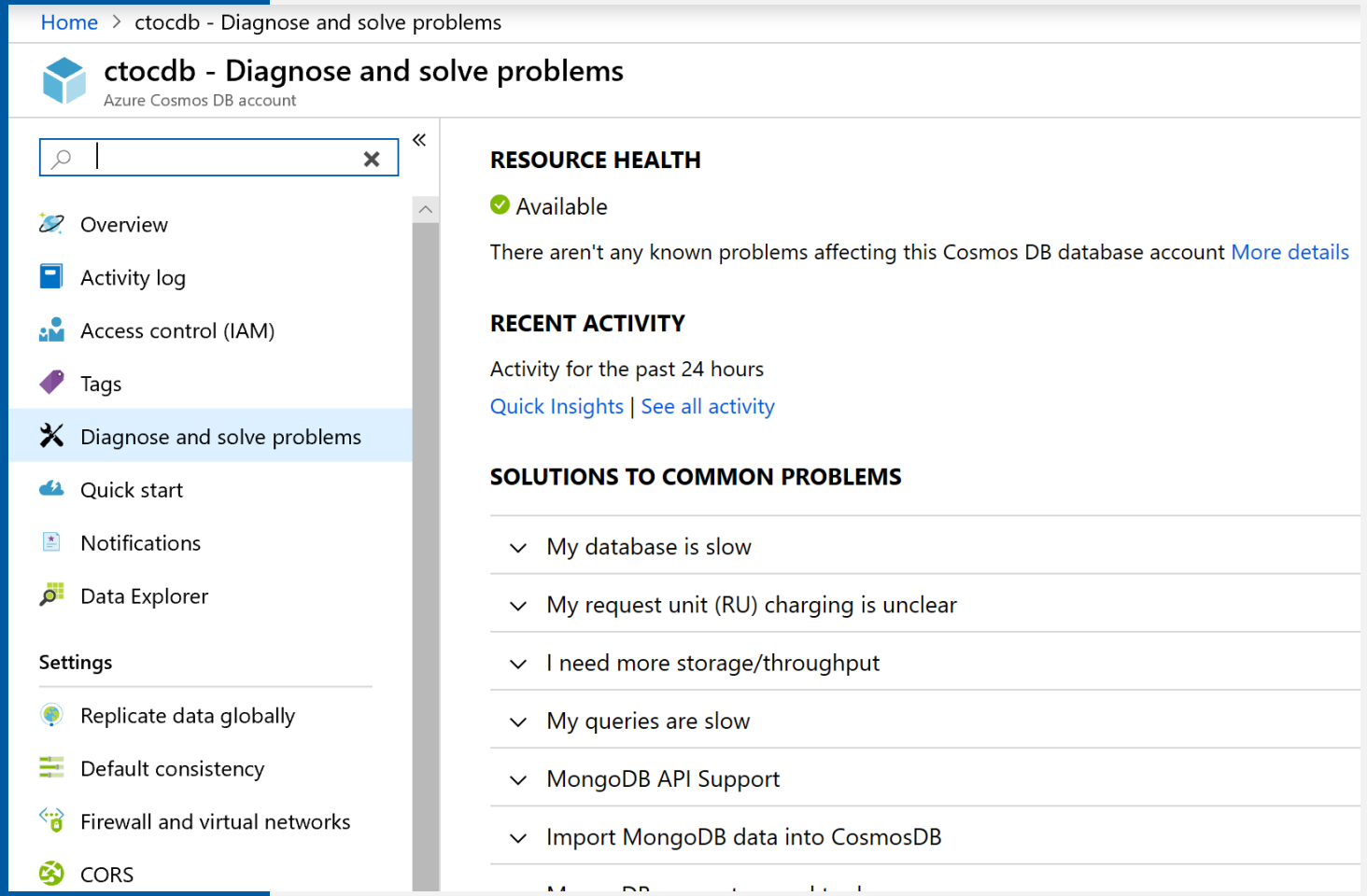
Network Performance Monitor measures the performance and reachability of the networks that you have configured.

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



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
• The first area that you should check is the firewall configuration. • Test the connection by accessing it from a location external to your network. • Check maintenance schedules	• The first check is to ensure that the user name and password is correct. • Check the storage account keys and ensure that they match in the connection string.	• Mongo client drivers establishes more than one connection. • On the server side, connections which are idle for more than 30 minutes are automatically closed down. • Check for timeouts	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.

Performance Issues

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:

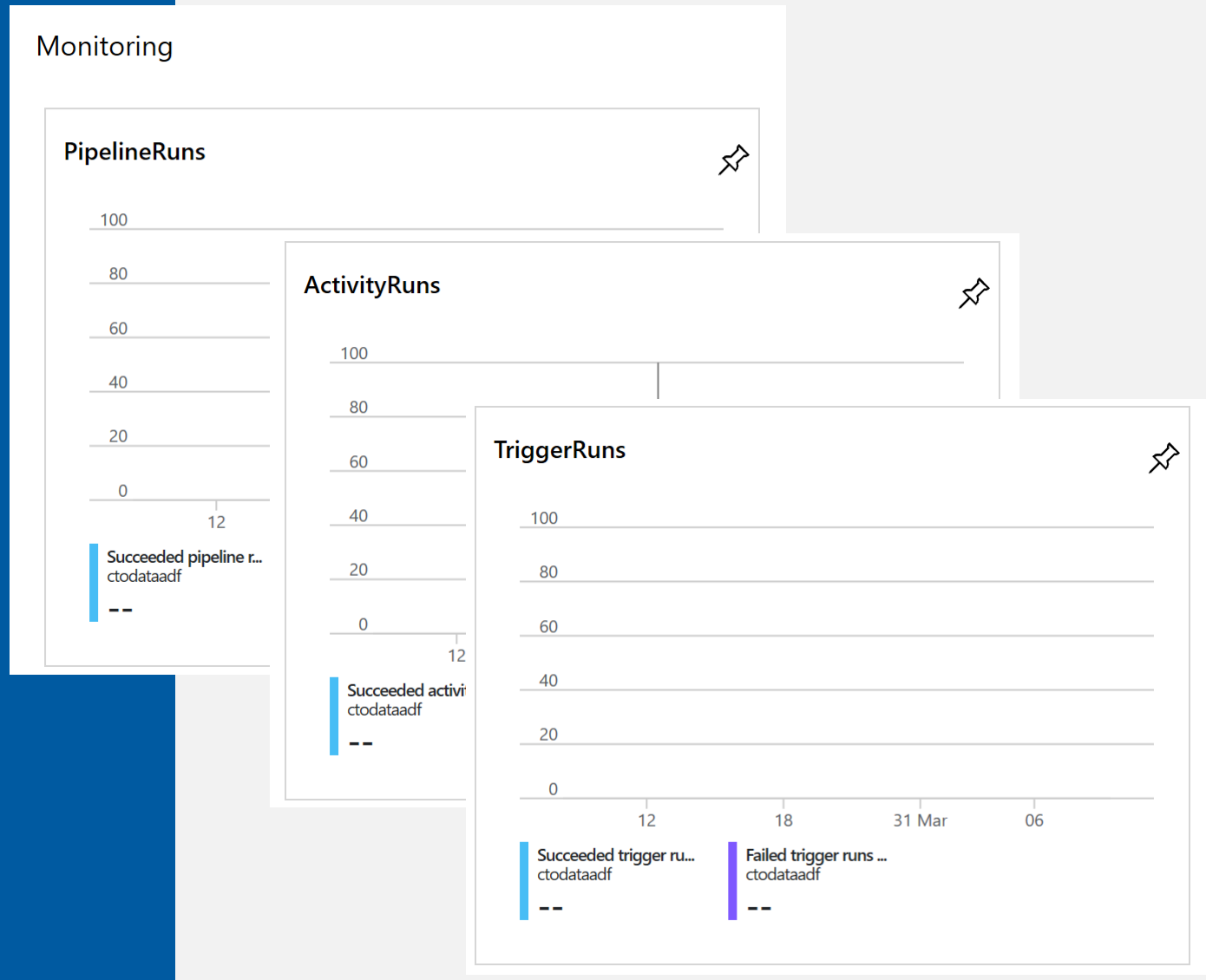
Job input	The job input contains a **Test Connection** button to validate that there is connectivity with the input. However, most errors associated with a job input is due to the malformed input data that is being ingested.
Job query	A common issue associated with Stream Analytics query is the fact 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



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 **geo-backup** 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 **geo-backup** 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 **4 hours** 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