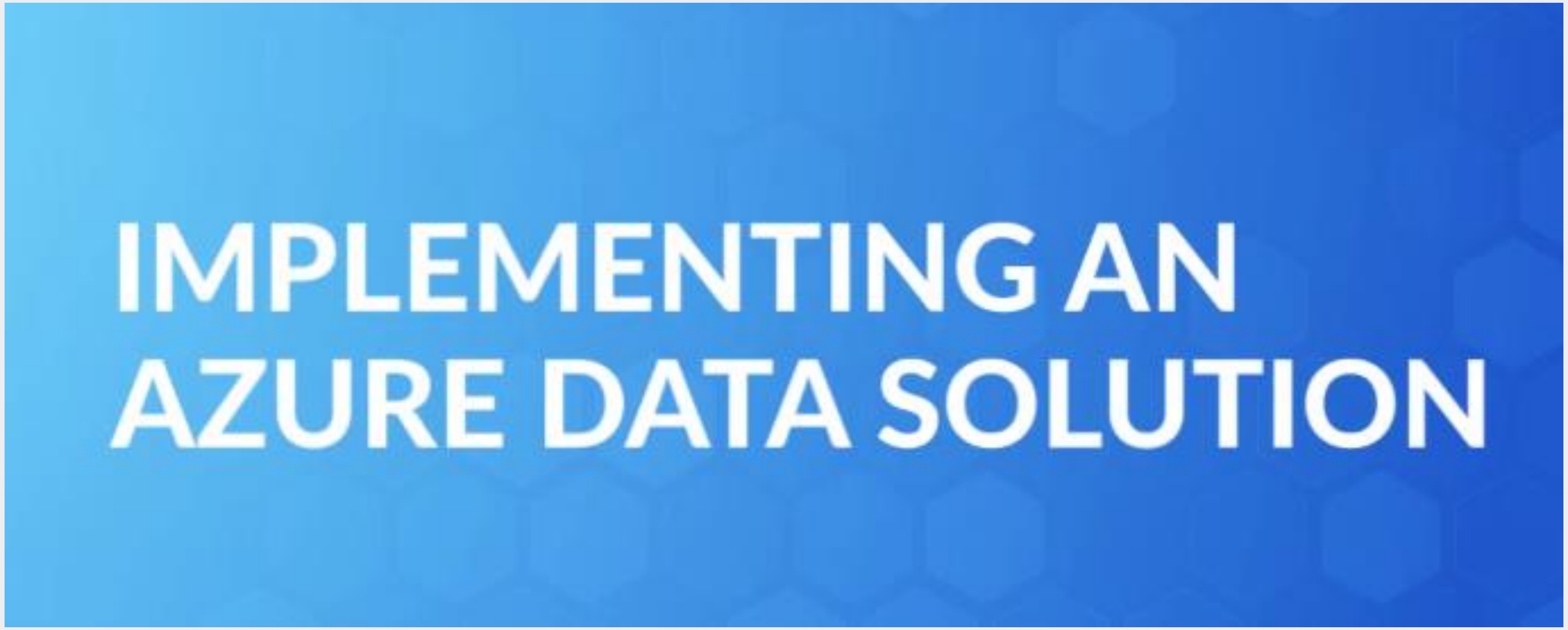


IMPLEMENTING AN AZURE DATA SOLUTION

IMPLEMENTING AN AZURE DATA SOLUTION

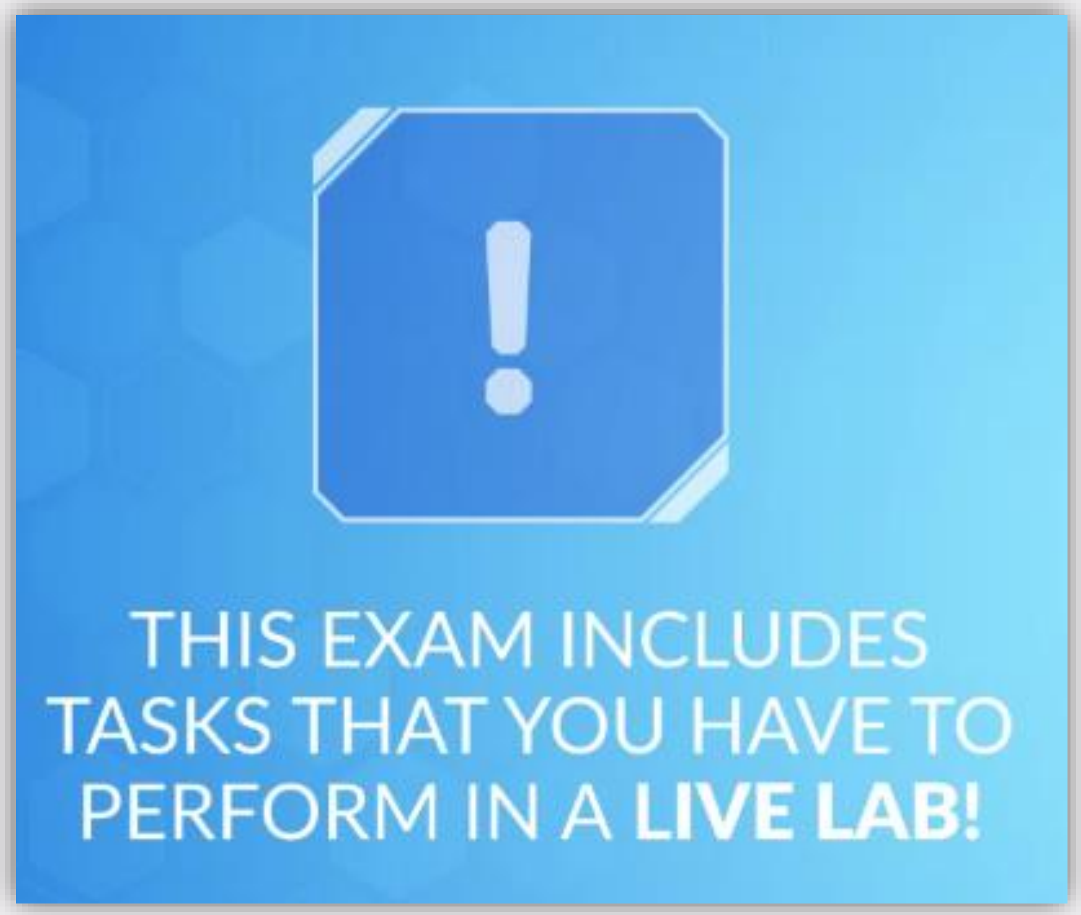
Microsoft Certified Azure Data Engineer Associate certification

THE DP-200 EXAM

Implementing data storage solutions

Managing and developing data processing

Monitoring and optimizing data solutions



THIS EXAM INCLUDES TASKS THAT YOU HAVE TO PERFORM IN A **LIVE LAB!**

Implementing an Azure Data Solution

Non-relational data stores

Relational data stores

Implementing an Azure Data Solution

Relational data stores

For many years, Microsoft's primary relational data solution was **SQL Server**

Azure SQL Database is usually the best option when migrating from an on-premises SQL Server to Azure

Relational data stores

Since **Azure SQL Database** is a managed service, it's easy to scale and provide high availability, disaster recovery, and global distribution

You will need to know how to configure all those features

If you really need full SQL Server compatibility, then you can use SQL Database Managed Instance

Implementing an Azure Data Solution

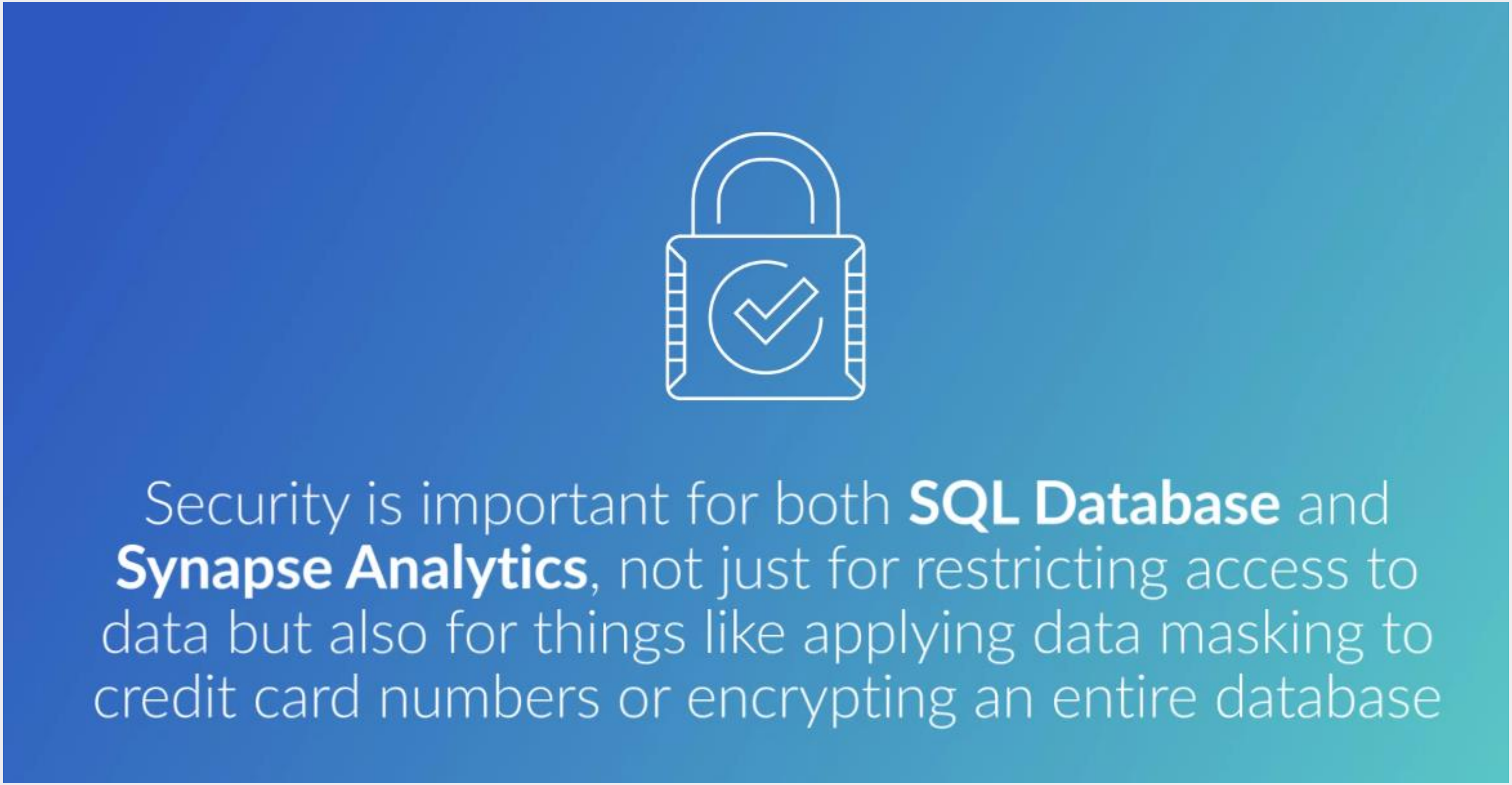
Relational data stores

Azure Synapse Analytics (formerly known as Azure SQL Data Warehouse)

It allows you to store and analyze huge amounts of data

Polybase is the fastest way to get data into Synapse Analytics

You need to partition the data store into multiple shards and use the right distribution method



Security is important for both **SQL Database** and **Synapse Analytics**, not just for restricting access to data but also for things like applying data masking to credit card numbers or encrypting an entire database

Implementing an Azure Data Solution

Non-relational data stores

These are services that can store unstructured data, such as documents or videos

Non-relational data stores

The most mature Azure service in this category is **Blob storage**

Unlike a filesystem, Blob storage has a flat structure

Non-relational data stores

For a true hierarchical structure, you can use **Azure Data Lake Storage Gen2**

It's especially useful for big data processing systems like **Azure Databricks**

Non-relational data stores

Cosmos DB can scale globally without sacrificing performance or flexibility

It supports multiple types of data models, including document, key-value, graph, and wide column

It supports five different consistency levels

As with **SQL Database** and **Synapse Analytics**, you need to know how to configure partitioning, security, high availability, disaster recovery, and global distribution for **Cosmos DB**

Implementing an Azure Data Solution

Managing and developing data processing solutions

Batch processing

Stream processing

Implementing an Azure Data Solution

Batch Processing

Azure Data Factory

Azure Databricks

Implementing an Azure Data Solution

Batch Processing

Data Factory makes it easy to copy data from one data store to another

It also makes it easy to transform data via services like Databricks

Implementing an Azure Data Solution

Batch Processing

Azure Databricks is a managed data analytics service based on Apache Spark

You can run Spark jobs on Azure HDInsight, but Databricks is the preferred solution

Some of the Databricks topics covered are data ingestion, clusters, notebooks, jobs, and autoscaling

Implementing an Azure Data Solution

Batch Processing

Azure Stream Analytics

You need to know how to:

- Get data into it from other services
- Process data streams using different windowing functions
- Output the results to another service

Monitoring and optimizing data solutions

Azure Monitor can be used to monitor and configure alerts for almost every other Azure service

One of the key components of Azure Monitor is **Log Analytics**, which you can use to implement auditing

Implementing an Azure Data Solution

The optimization subsection

You need to know how to optimize the performance of services like:

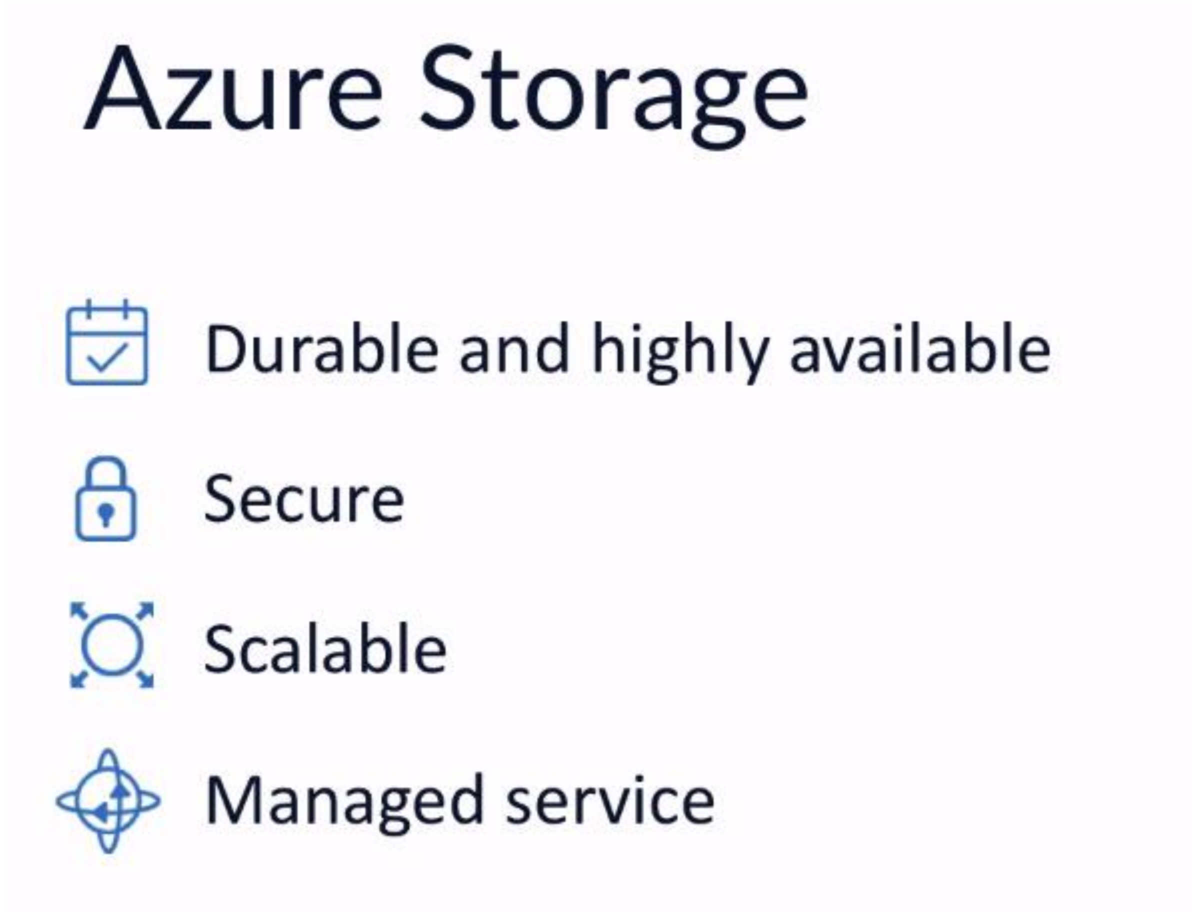
- Stream Analytics
- SQL Database
- Synapse Analytics

Using the right partitioning method is one of the most important optimization techniques

This learning path assumes that you already have some basic experience using **Microsoft Azure**

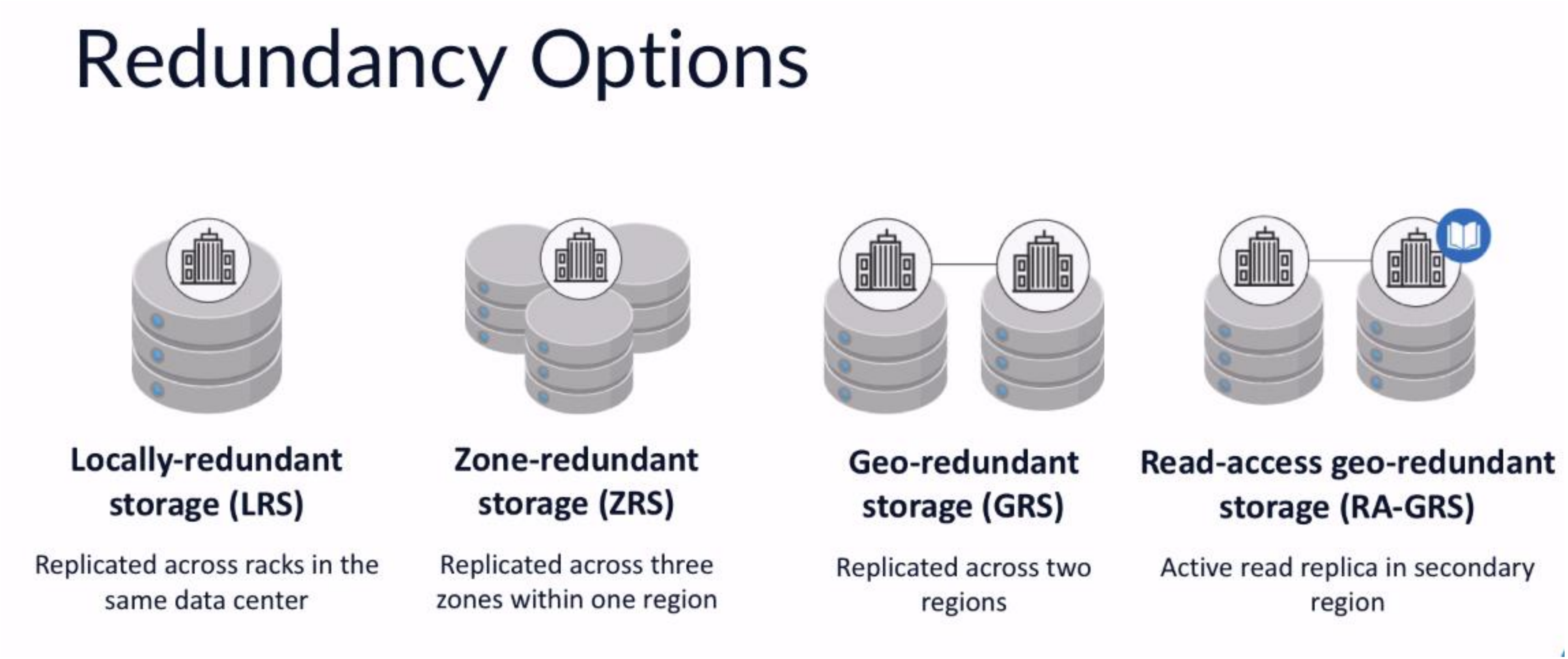
Working with Data Storage

Azure Storage



Azure Storage

- Durable and highly available
- Secure
- Scalable
- Managed service



Redundancy Options

Locally-redundant storage (LRS)

Replicated across racks in the same data center

Zone-redundant storage (ZRS)

Replicated across three zones within one region

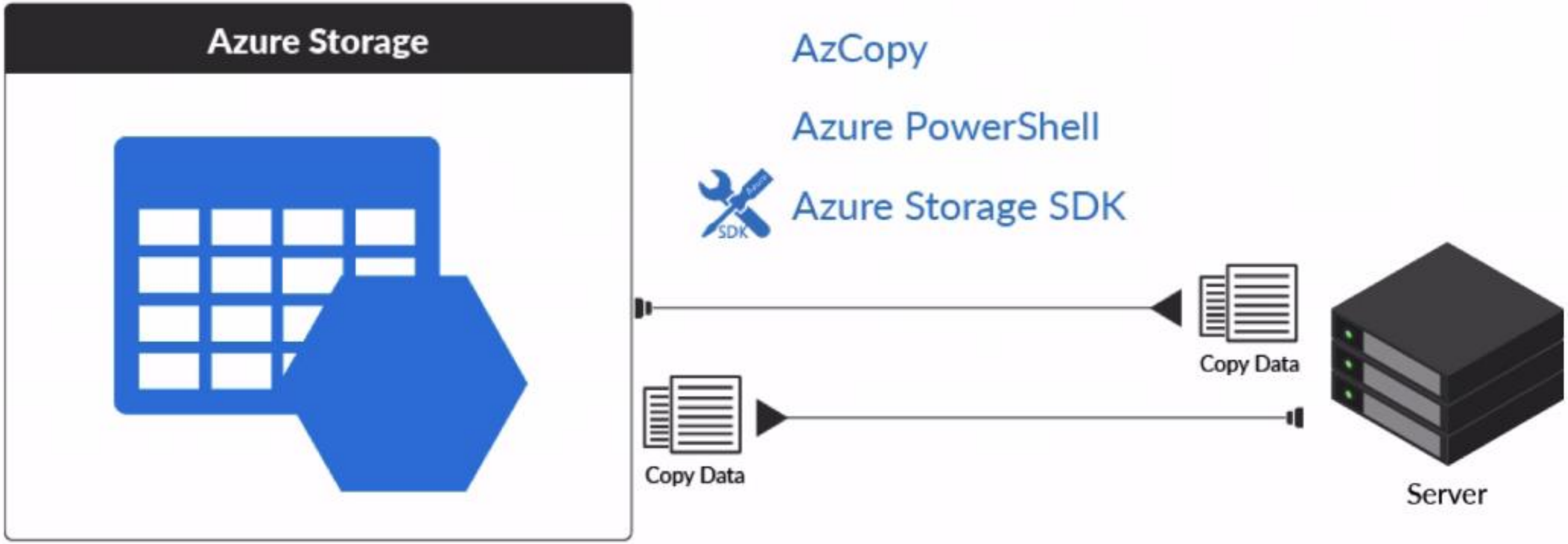
Geo-redundant storage (GRS)

Replicated across two regions

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

Active read replica in secondary region

Tools for Copying Data



Azure Storage

AzCopy

Azure PowerShell

Azure Storage SDK

Copy Data

Copy Data

Server

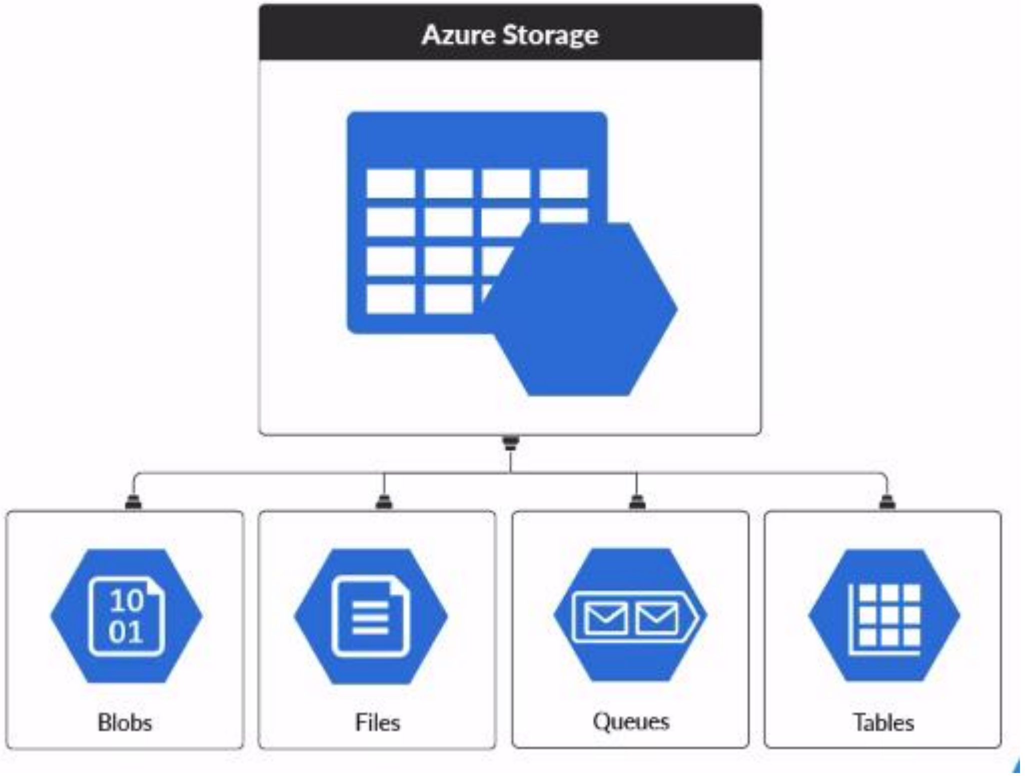
Data Types

Blobs

- Binary large object
- No organizational structure

Files

- Filesystem structure
- SMB-compliant
- Move on-premises file share to Azure
- Accessible over the web - need a shared access signature token
- Significantly more expensive than Blob storage



Blob Storage Tiers

Hot

- For data that gets accessed frequently

Cool

- For data that doesn’t get accessed frequently
- Data gets retrieved immediately
- Lower storage cost, higher cost for reads and writes
- 30-day minimum

Blob Storage Tiers

Hot

- For data that gets accessed frequently

Cool

- For data that doesn’t get accessed frequently
- Data gets retrieved immediately
- Lower storage cost, higher cost for reads and writes
- 30-day minimum

Archive

- Takes up to 15 hours to access when requested
- 5 times cheaper than cool tier, but far more expensive for reads
- 180-day minimum

Data

Moving data from cool or archive tiers before minimum duration incurs an early deletion fee

Data Types

Queues

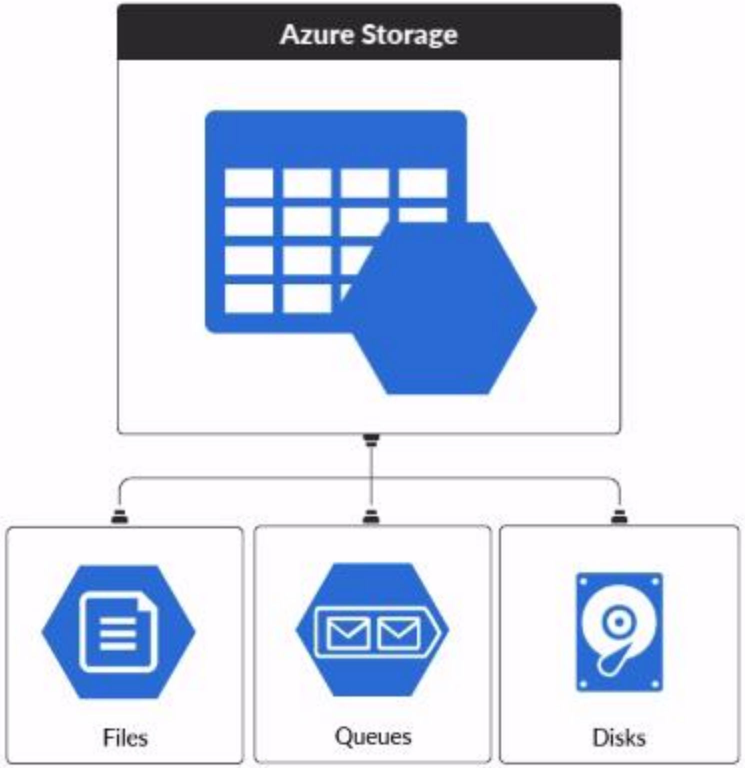
- For passing messages
- One application pushes messages onto queue, another retrieves them

Tables

- NoSQL datastore
- Storage costs about the same as File storage, but much lower transaction costs
- Premium version is part of CosmosDB service

Disks

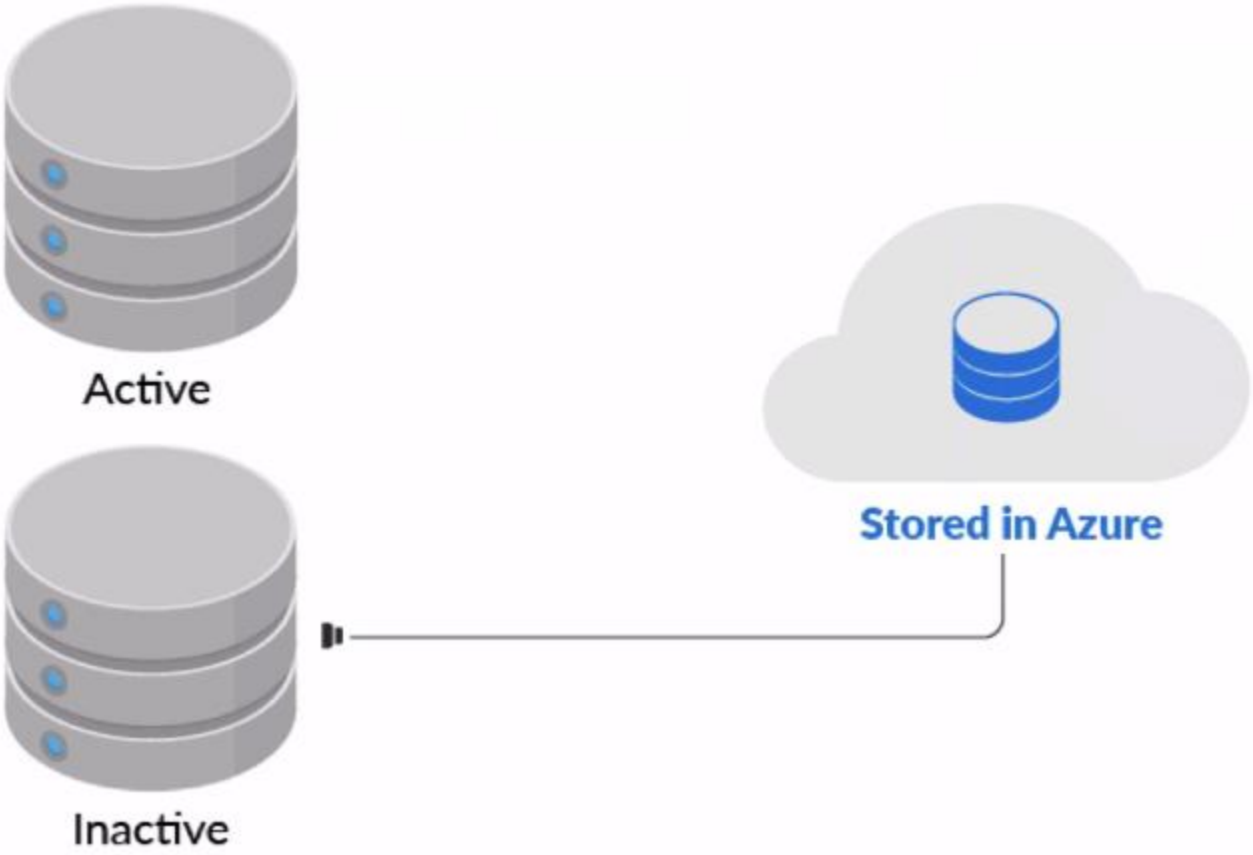
- Attached to virtual machines



StorSimple

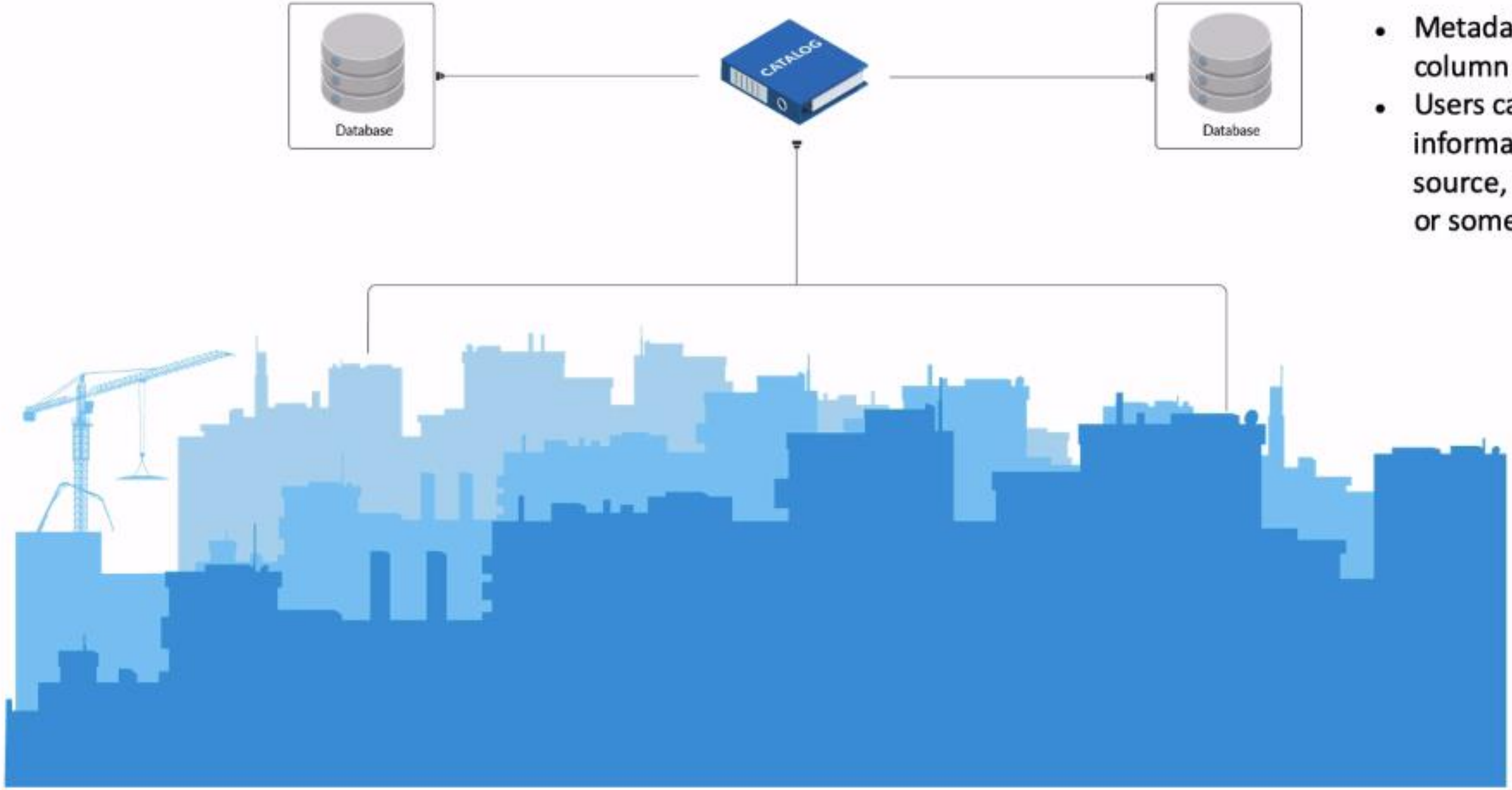
- Virtual array installed on-premises
- Backup
- Recovery
- Storage tiering

StorSimple



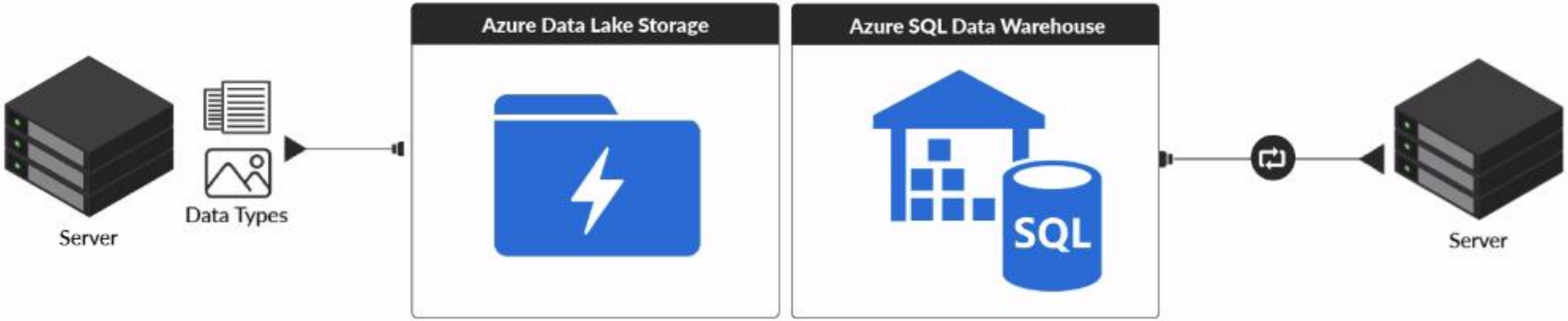
Azure Data Services

Azure Data Catalog



- Metadata includes things like column names and data types
- Users can also add additional information about a data source, such as a description or some tags

Azure Data Lake and Data Warehouse



If you have raw data that's not in a nicely structured format, then you'll probably need to process it before you store it.

Azure Data Lake and Data Warehouse

Azure Data Lake Storage

Built to work with Hadoop
No regulatory compliance
Write queries using U-SQL

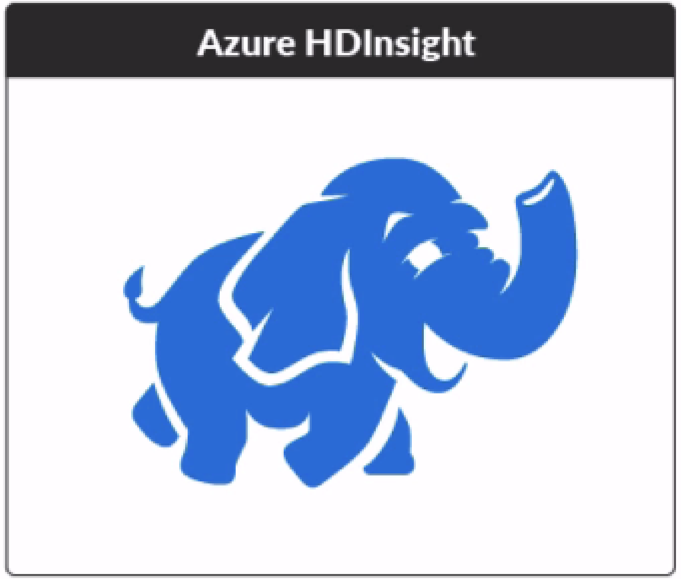
Azure SQL Data Warehouse

Built on SQL Server
Certified for compliance
Write queries using T-SQL

Azure HDInsight

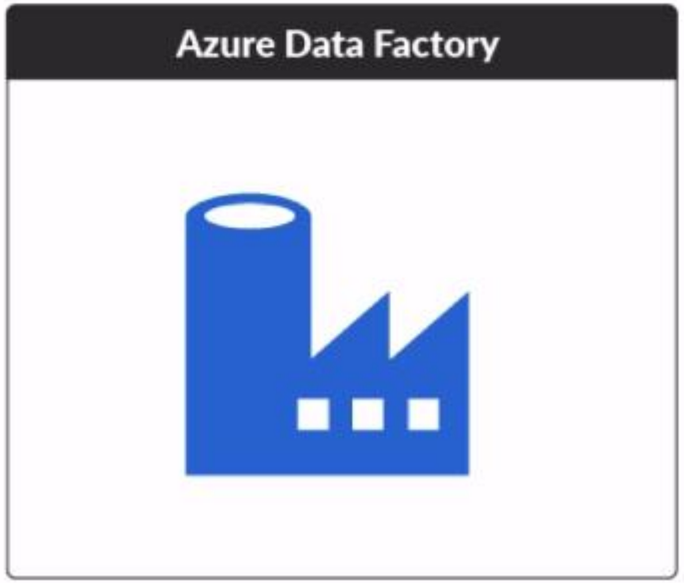
Supports a wide variety of open-source big data frameworks, including:

- Hadoop
- Spark
- Hive
- Storm
- Many others



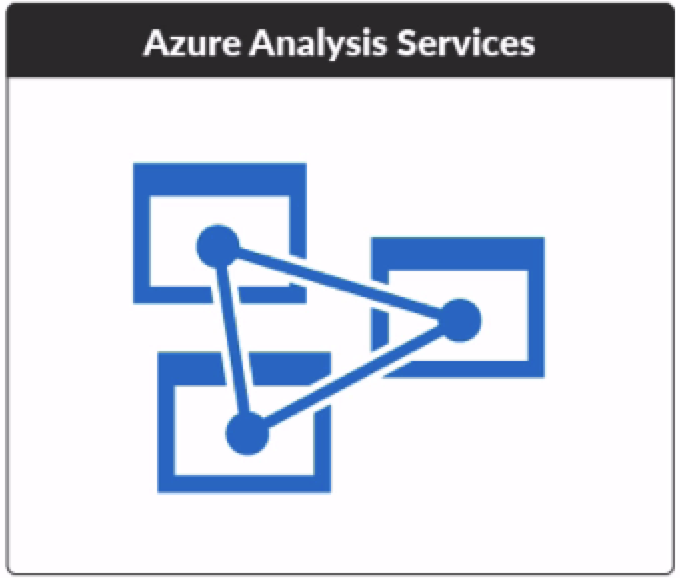
Azure Data Factory

- Automates data movement and data transformation
- Spins up and down HDInsight clusters as needed
- Creates data processing pipelines
- Automates Data Lake Analytics queries and machine learning



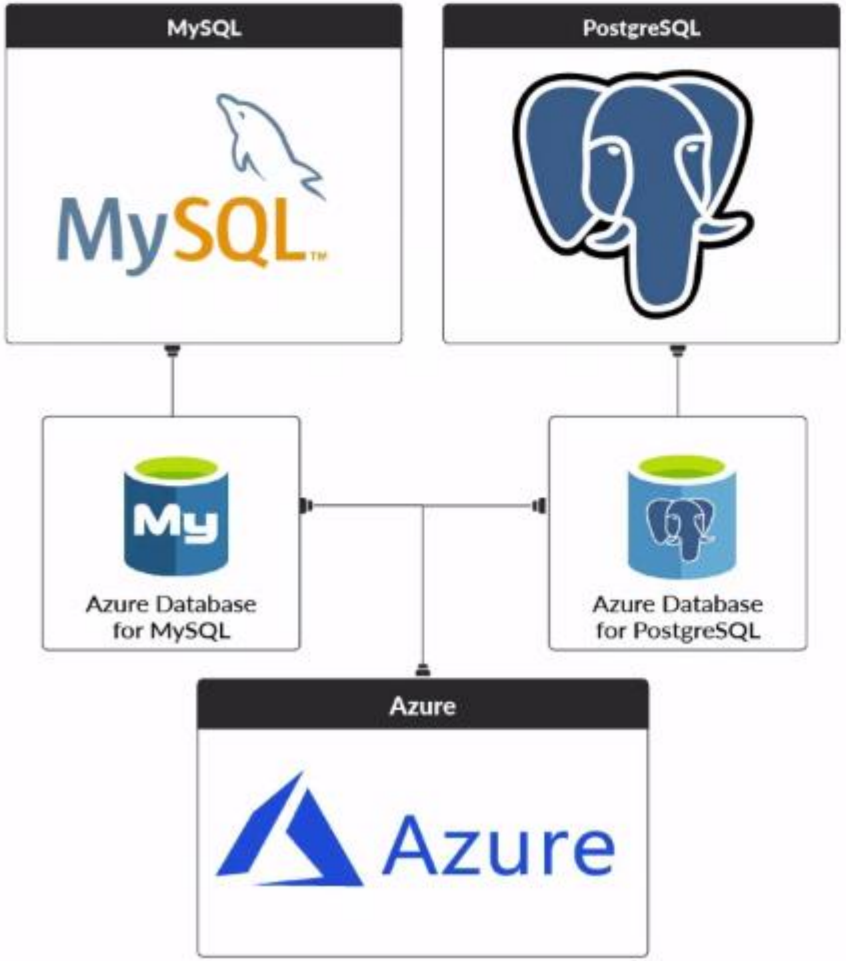
Azure Analysis Services

- Lets you create data models of existing data
- Uses in-memory caching
- Accessed through supported client tools such as Power BI, Tableau or Excel



Relational Database Storage

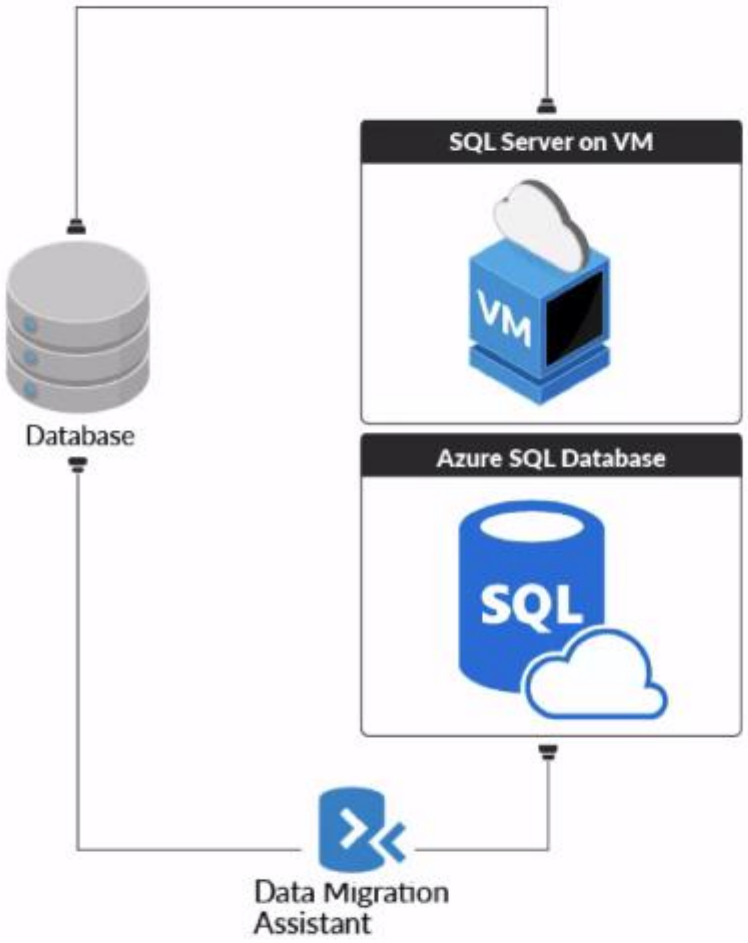
Relational Databases



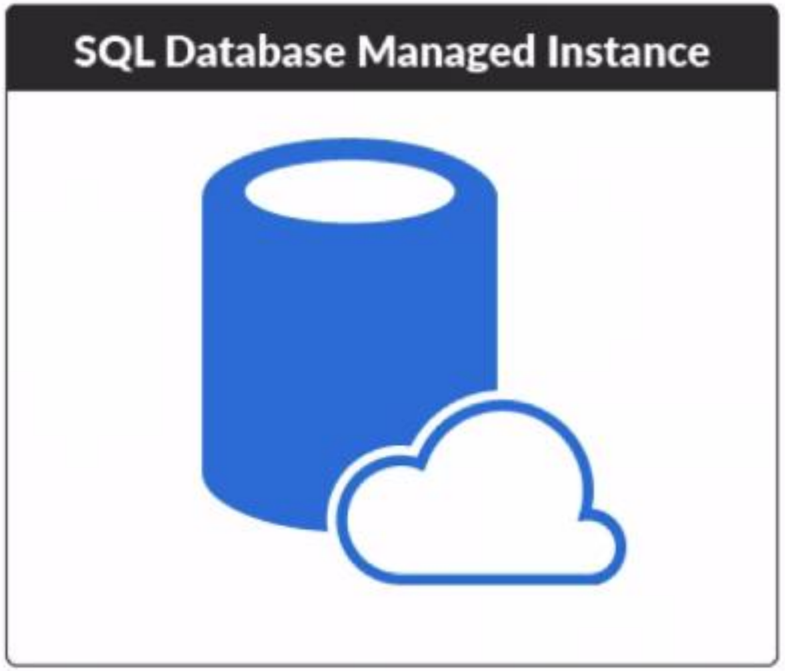
SQL Server Stretch Database

- Migrates cold table rows to Azure
- You can still query migrated data
- Avoid buying more storage
- Shorten backup times
- More expensive than storing data offline
- Data Migration Assistant
  - Tells you which tables would be good candidates for Stretch Database
  - Indicates potential blocking issues

Moving the Entire Database to Azure



SQL Database Managed Instance



Nearly **100%** compatible with SQL Server

Azure SQL Database Service Tiers

General Purpose

Hyperscale

Business Critical

Azure SQL Database Service Tiers

General Purpose

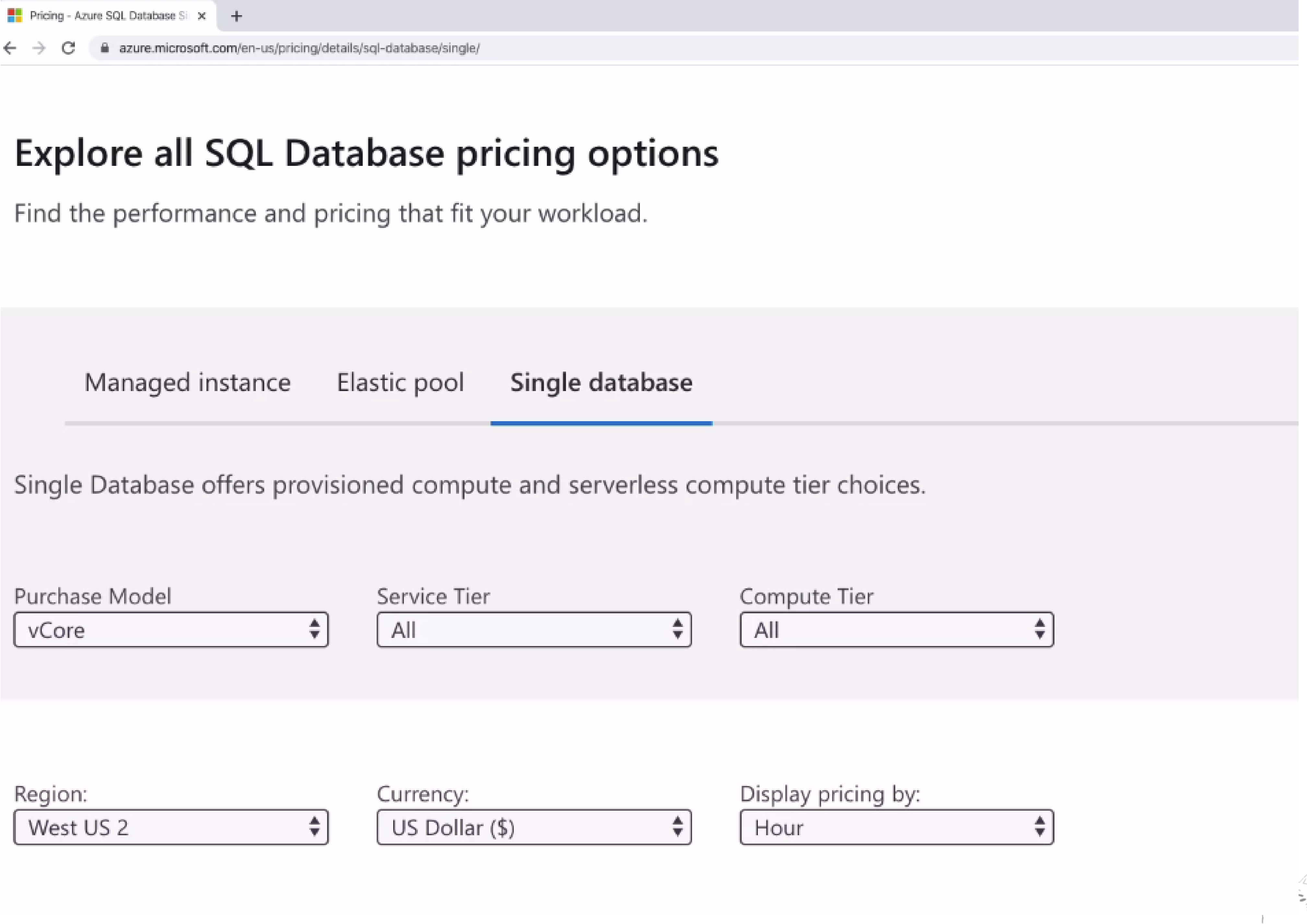
- Least expensive
- Latency: 5 - 10 milliseconds
- Availability: 99.99%
- Max size: 4TB (8TB for Managed Instance)

Hyperscale

- Max size: 100TB
- Scales compute resources up and down very quickly
- Instant backups and fast database restores

Business Critical

- Latency: 1 - 2 milliseconds
- Local SSDs on 4-node cluster
- Most expensive
- Availability: 99.995% (with zone-redundant option)
- Max size: 4TB


Pricing - Azure SQL Database Si
azure.microsoft.com/en-us/pricing/details/sql-database/single/

Explore all SQL Database pricing options

Find the performance and pricing that fit your workload.

Managed instance
Elastic pool
Single database

Single Database offers provisioned compute and serverless compute tier choices.

Purchase Model: vCore

Service Tier: All

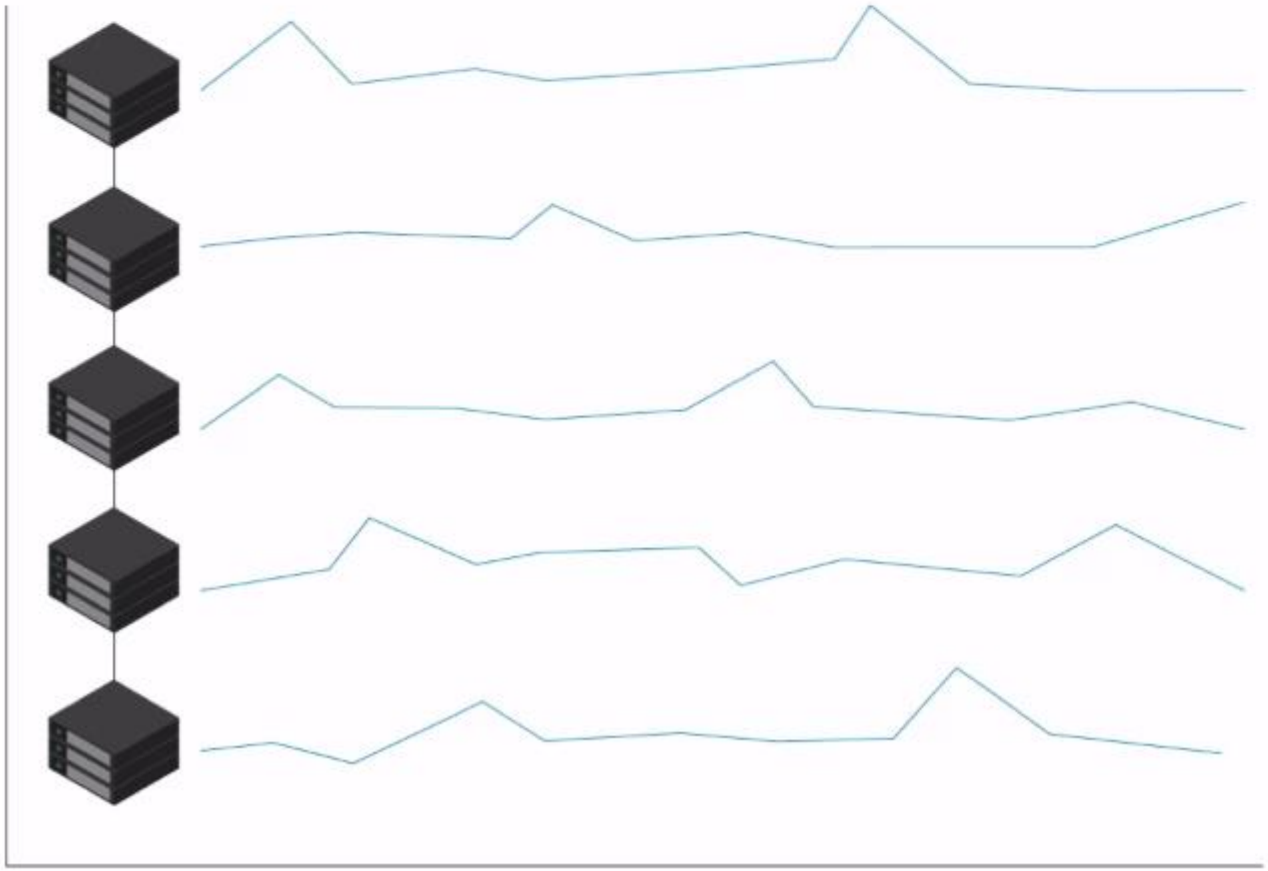
Compute Tier: All

Region: West US 2

Currency: US Dollar ($)

Display pricing by: Hour

Elastic Pool Model



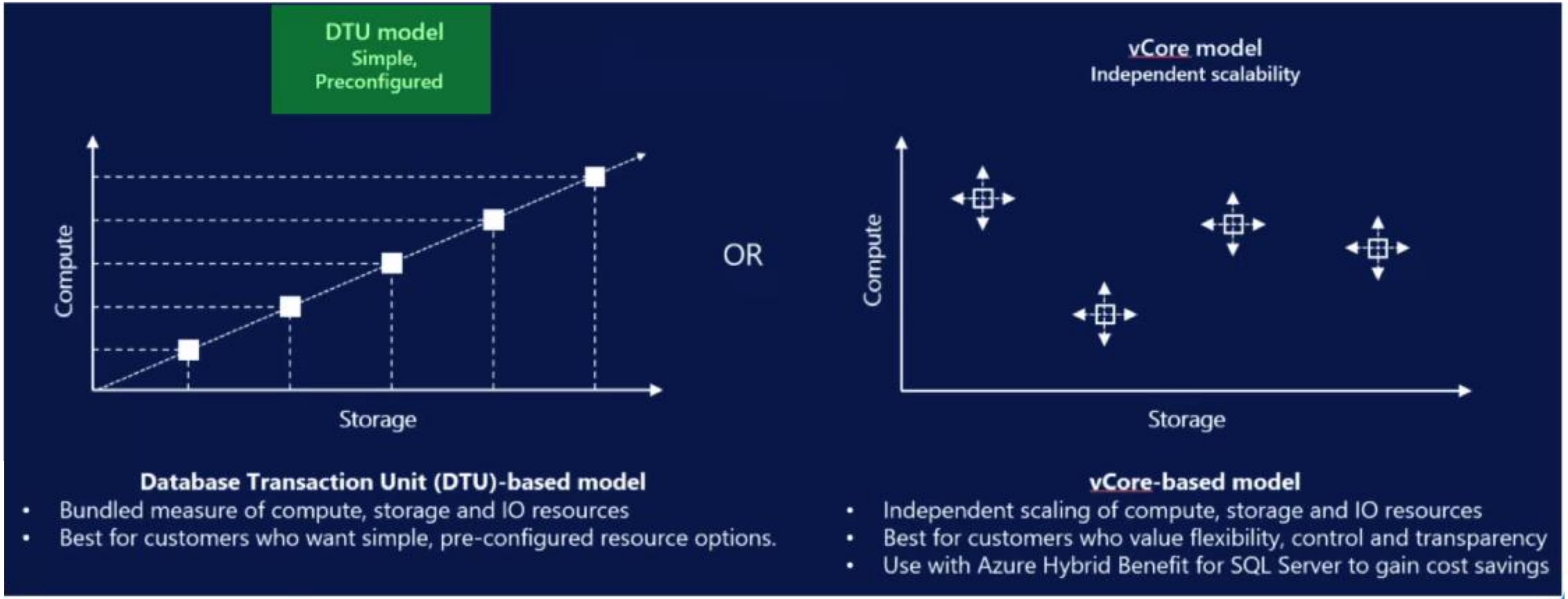
SQL Database Managed Instance

Nearly 100% compatible with SQL Server

Doesn’t work with the Hyperscale tier

(Elastic pools don’t work with the Hyperscale tier either)

Purchasing Models



Database Transaction Unit (DTU)-based model

- Bundled measure of compute, storage and IO resources
- Best for customers who want simple, pre-configured resource options.

vCore-based model

- Independent scaling of compute, storage and IO resources
- Best for customers who value flexibility, control and transparency
- Use with Azure Hybrid Benefit for SQL Server to gain cost savings

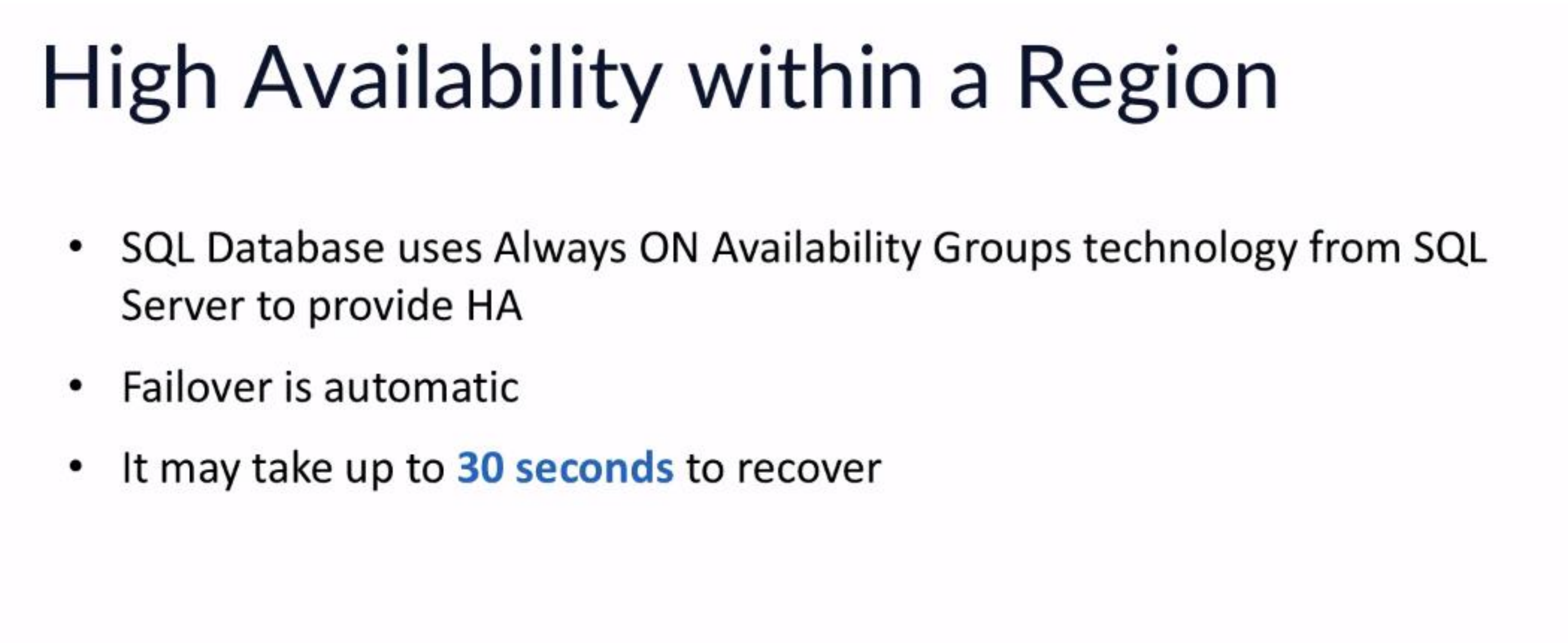
vCore Compute Tiers

Provisioned

- Provisioned with exact resources requested
- You get charged for the database as long it's running
- To scale manually, change the number of vCores—there will be a brief loss of connectivity (<4 sec.)

Serverless

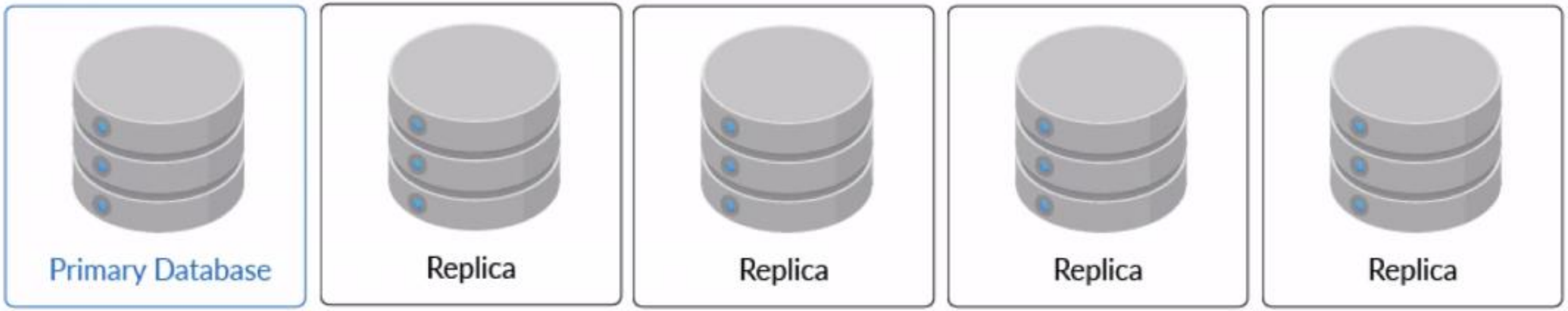
- Minimum and maximum vCores
- Autoscales based on workload demand
- If no activity, it pauses database and halts compute charges
- Cost per vCore is higher



High Availability within a Region

- SQL Database uses Always ON Availability Groups technology from SQL Server to provide HA
- Failover is automatic
- It may take up to **30 seconds** to recover

Active Geo-Replication



Primary Database

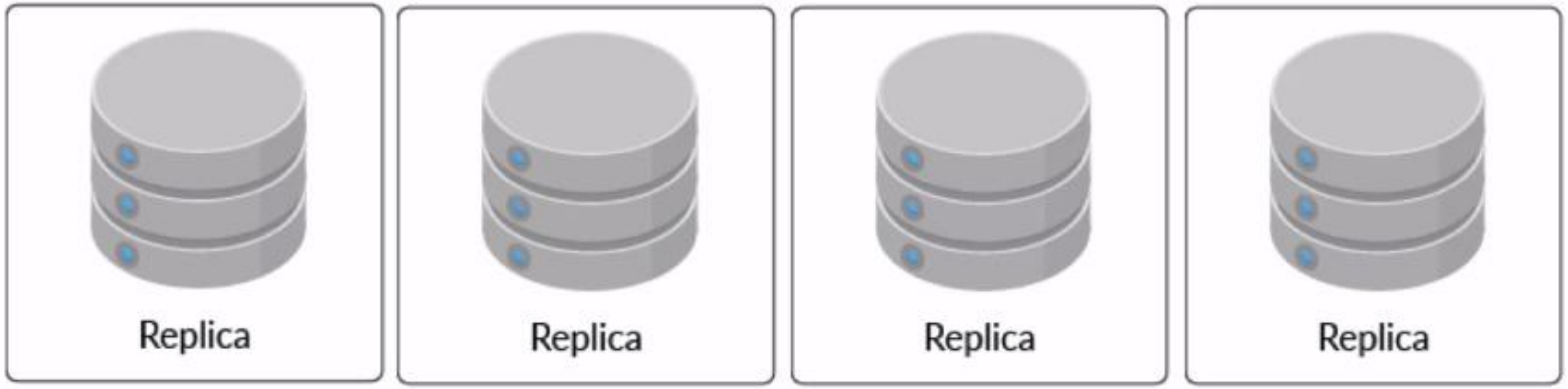
Replica

Replica

Replica

Replica

Active Geo-Replication



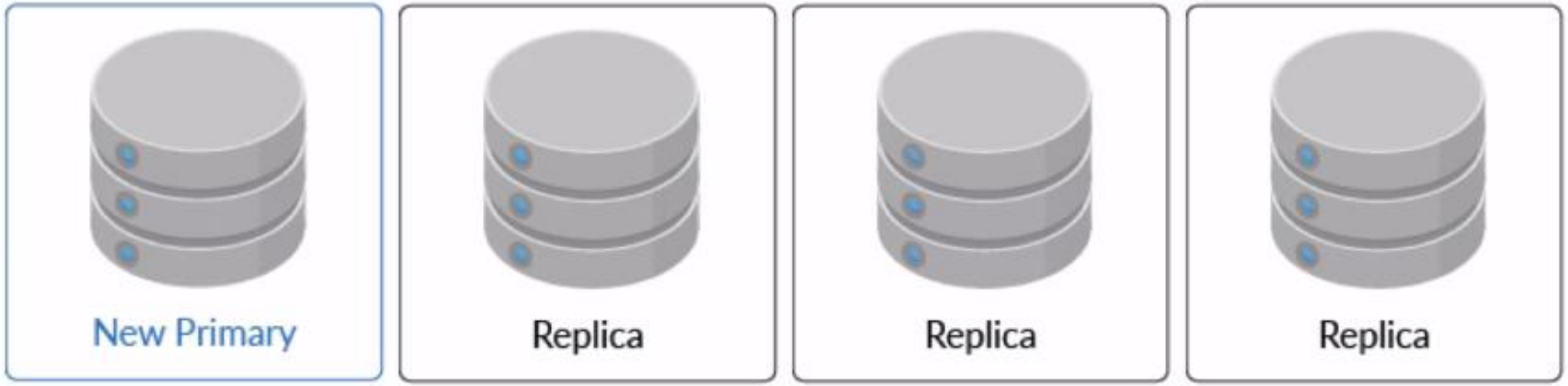
Replica

Replica

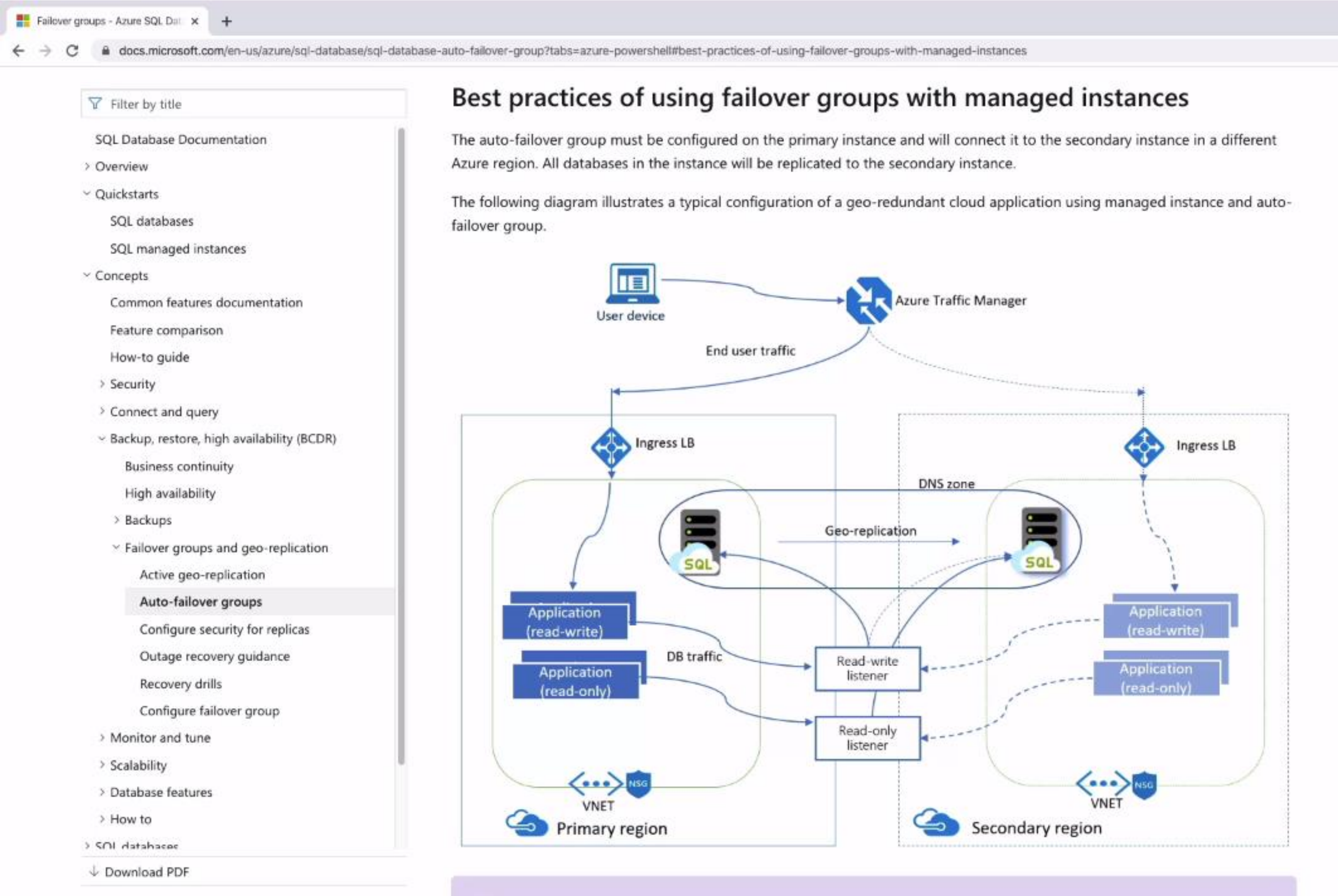
Replica

Replica

Active Geo-Replication



- Ensure secondaries have the same user authentication configuration as the primary
- You should use the same firewall rules for secondaries as you do for the primary
- Active geo-replication allows you to use the secondary databases to make queries faster for users in other regions



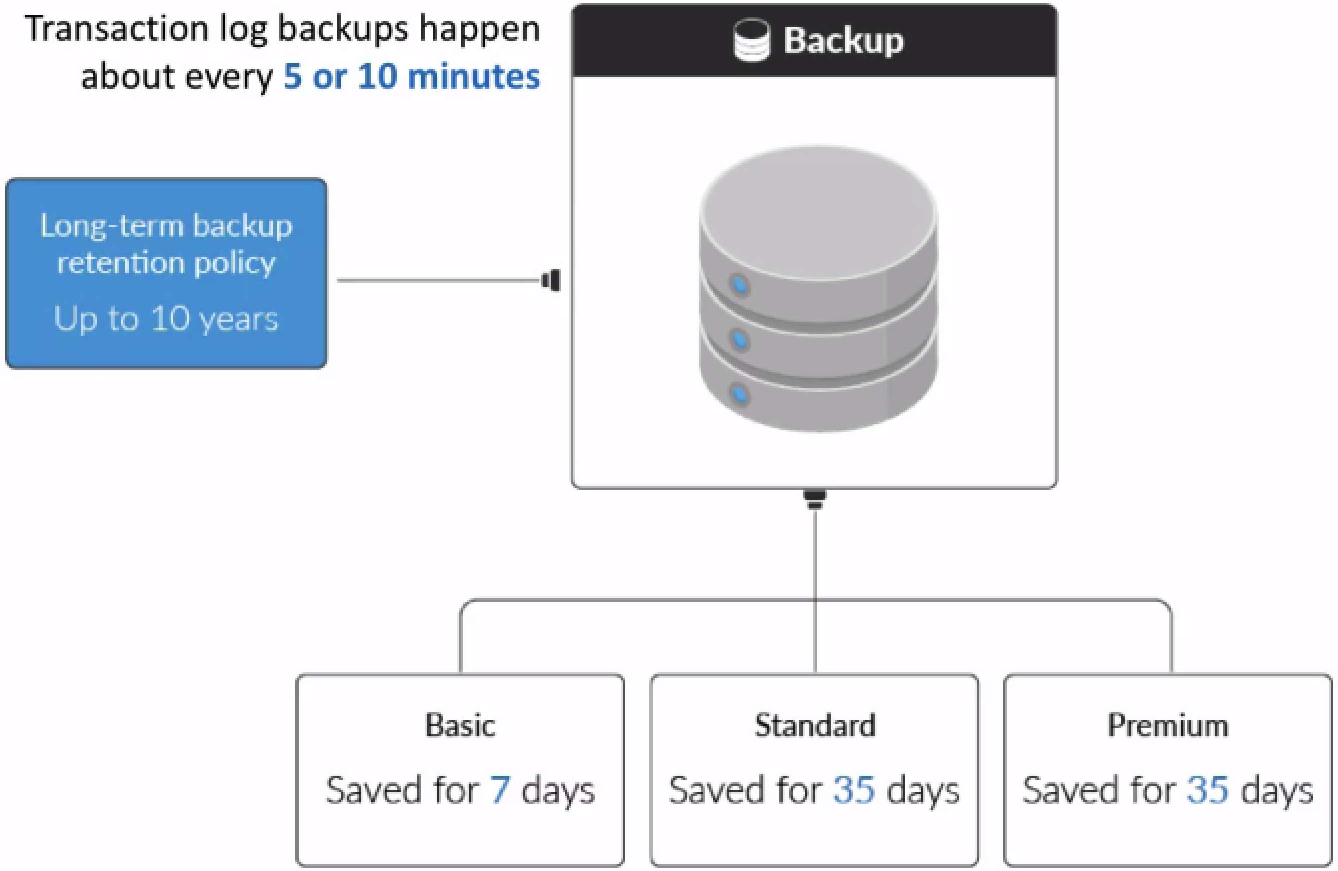
Failover groups - Azure SQL Dat
docs.microsoft.com/en-us/azure/sql-database/sql-database-auto-failover-group?tabs=azure-powershell#best-practices-of-using-failover-groups-with-managed-instances
Filter by title
SQL Database Documentation
Overview
Quickstarts
SQL databases
SQL managed instances
Concepts
Common features documentation
Feature comparison
How-to guide
Security
Connect and query
Backup, restore, high availability (BCDR)
Business continuity
High availability
Backups
Failover groups and geo-replication
Active geo-replication
Auto-failover groups
Configure security for replicas
Outage recovery guidance
Recovery drills
Configure failover group
Monitor and tune
Scalability
Database features
How to
SQL databases
Download PDF

Best practices of using failover groups with managed instances

The auto-failover group must be configured on the primary instance and will connect it to the secondary instance in a different Azure region. All databases in the instance will be replicated to the secondary instance.

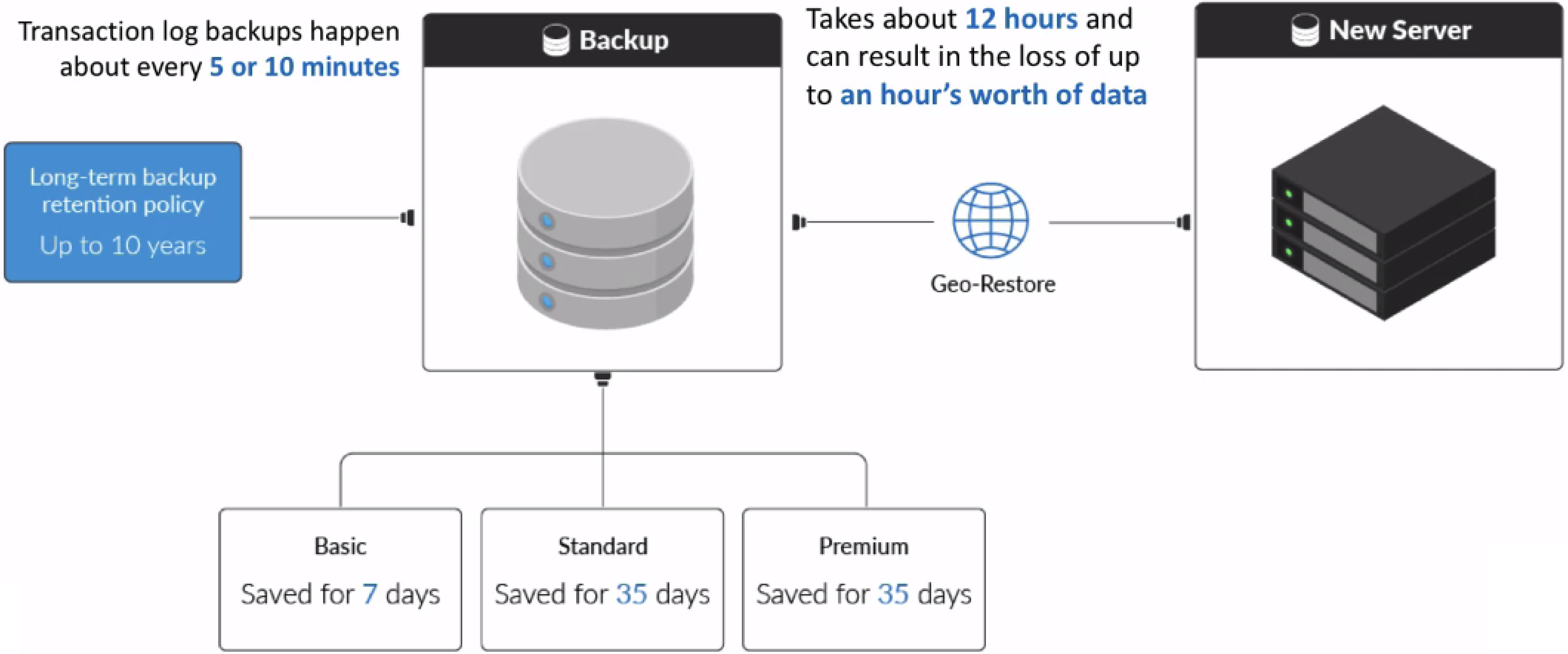
The following diagram illustrates a typical configuration of a geo-redundant cloud application using managed instance and auto-failover group.

Backups



Transaction log backups happen about every **5 or 10 minutes**

Backups



Transaction log backups happen about every **5 or 10 minutes**

Long-term backup retention policy

Up to 10 years

Backup

Takes about **12 hours** and can result in the loss of up to **an hour's worth of data**

Geo-Restore

New Server

Basic

Saved for 7 days

Standard

Saved for 35 days

Premium

Saved for 35 days

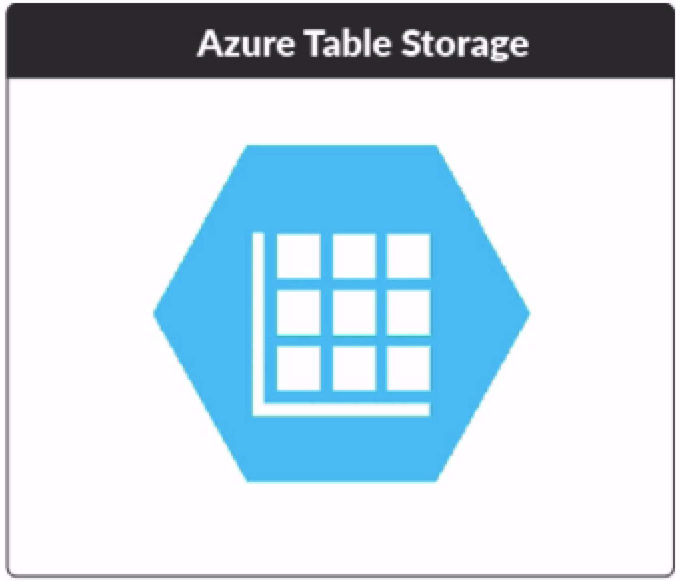
NoSQL Storage

NoSQL Datastores

- Scale better
- Satisfy fewer requirements than relational databases

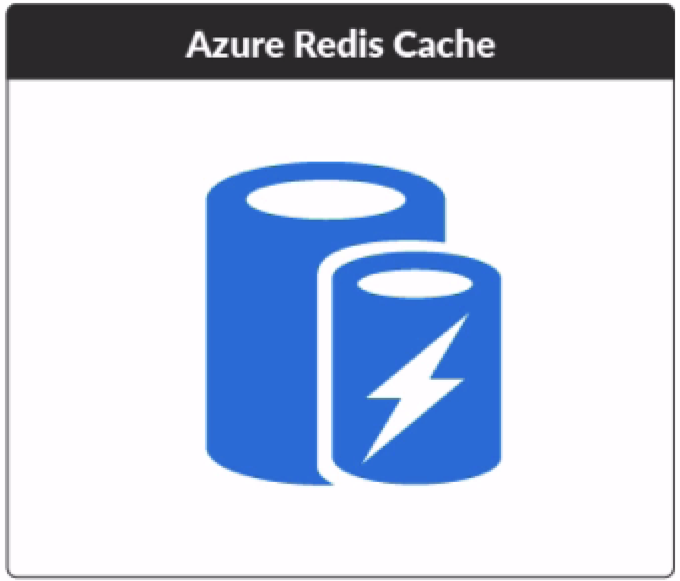
Azure Table Storage

- Intended for simple structured data
- E.g. address books and user profiles
- Schemaless design
- Indexes records
- For secondary indexes or global distribution, use Cosmos DB version instead
- For complex joins, foreign keys, or stored procedures, use relational database instead



Azure Redis Cache

- Intended to speed up data retrieval in applications
- Managed service for Redis
- Data resides in memory
- It stores key/value pairs



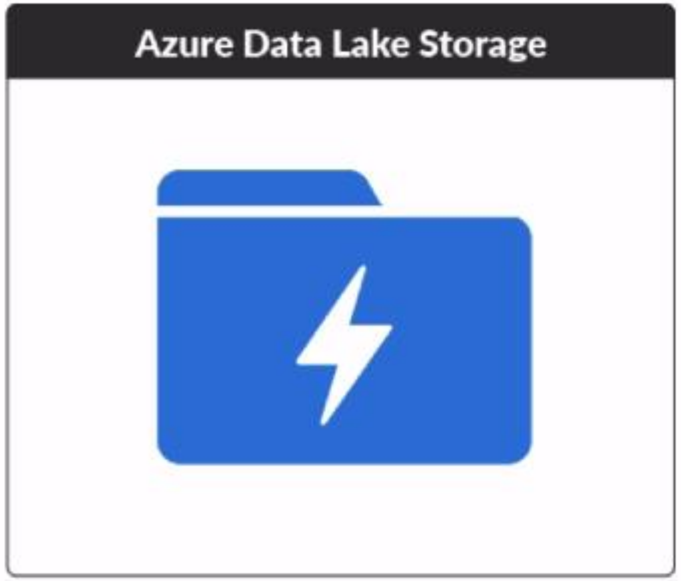
Basic: Should only be used for testing and development

Standard: Provides a replicated, high availability cache

Premium: Better performance and can handle bigger workloads, disaster recovery, and more

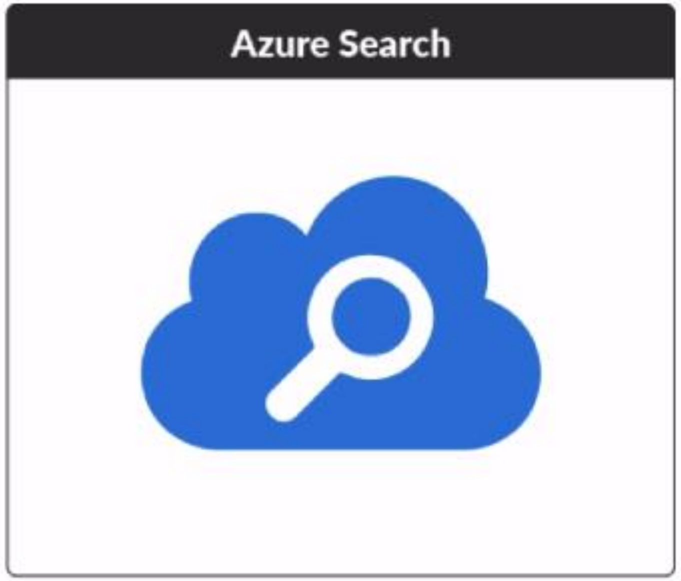
Azure Data Lake Storage

- Intended to hold large quantities of any kind of data
- Data warehouse for unstructured data
- Main purpose is data analytics



Azure Search

- Creates an index of text data
- You can embed search functionality into web, mobile, and enterprise applications
- Offers features such as:
  - Search suggestions
  - Language analyzers
  - Fuzzy searches

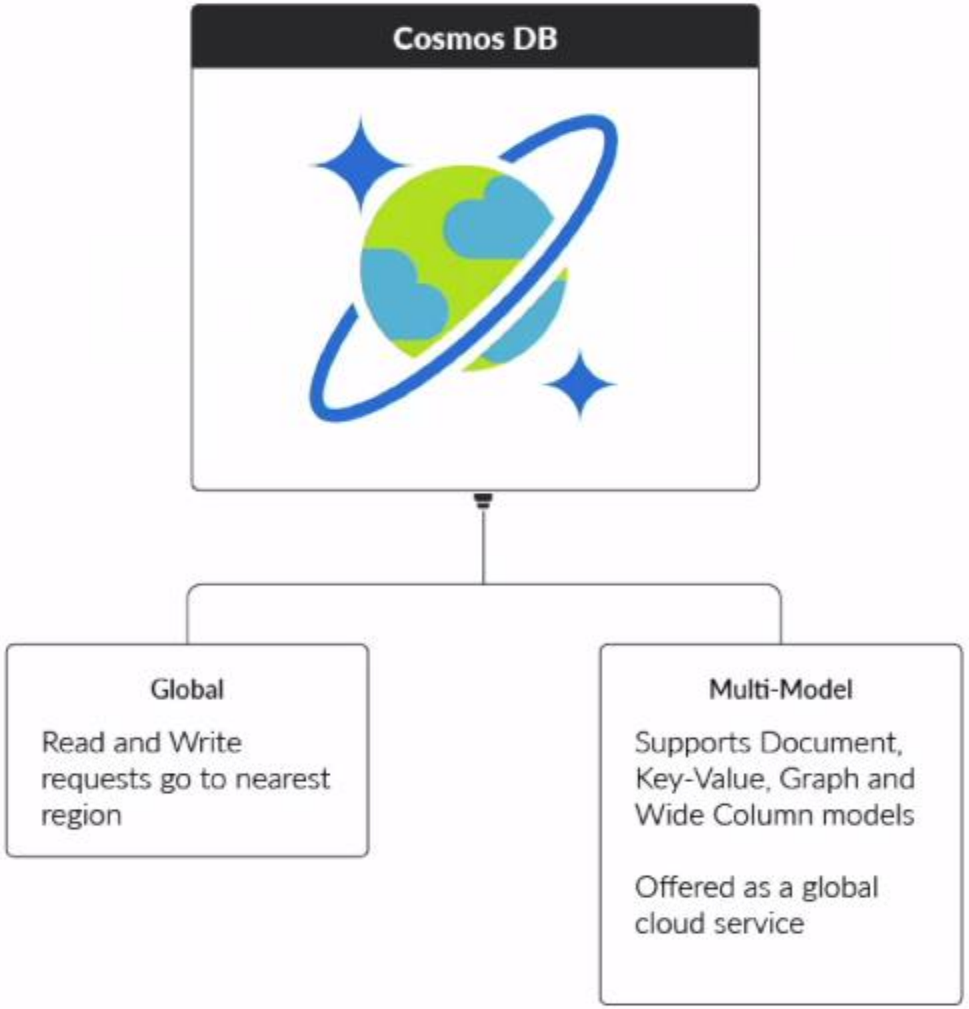


Time Series Insights (TSI)

- Collects time-stamped data
- Integrates with Azure IoT Hub and Azure Events Hubs
- Run queries on billions of events and get a response in seconds
- See visualizations of the data with TSI Explorer

Cosmos DB

Introduction to Cosmos DB



Cosmos DB

Global

Read and Write requests go to nearest region

Multi-Model

Supports Document, Key-Value, Graph and Wide Column models

Offered as a global cloud service

Table API

Cosmos DB’s Table API is built on Table storage, but offers these additional features:

- Global distribution
- Dedicated throughput worldwide
- Single-digit millisecond latencies at the 99th percentile
- Guaranteed high availability
- Automatic secondary indexing

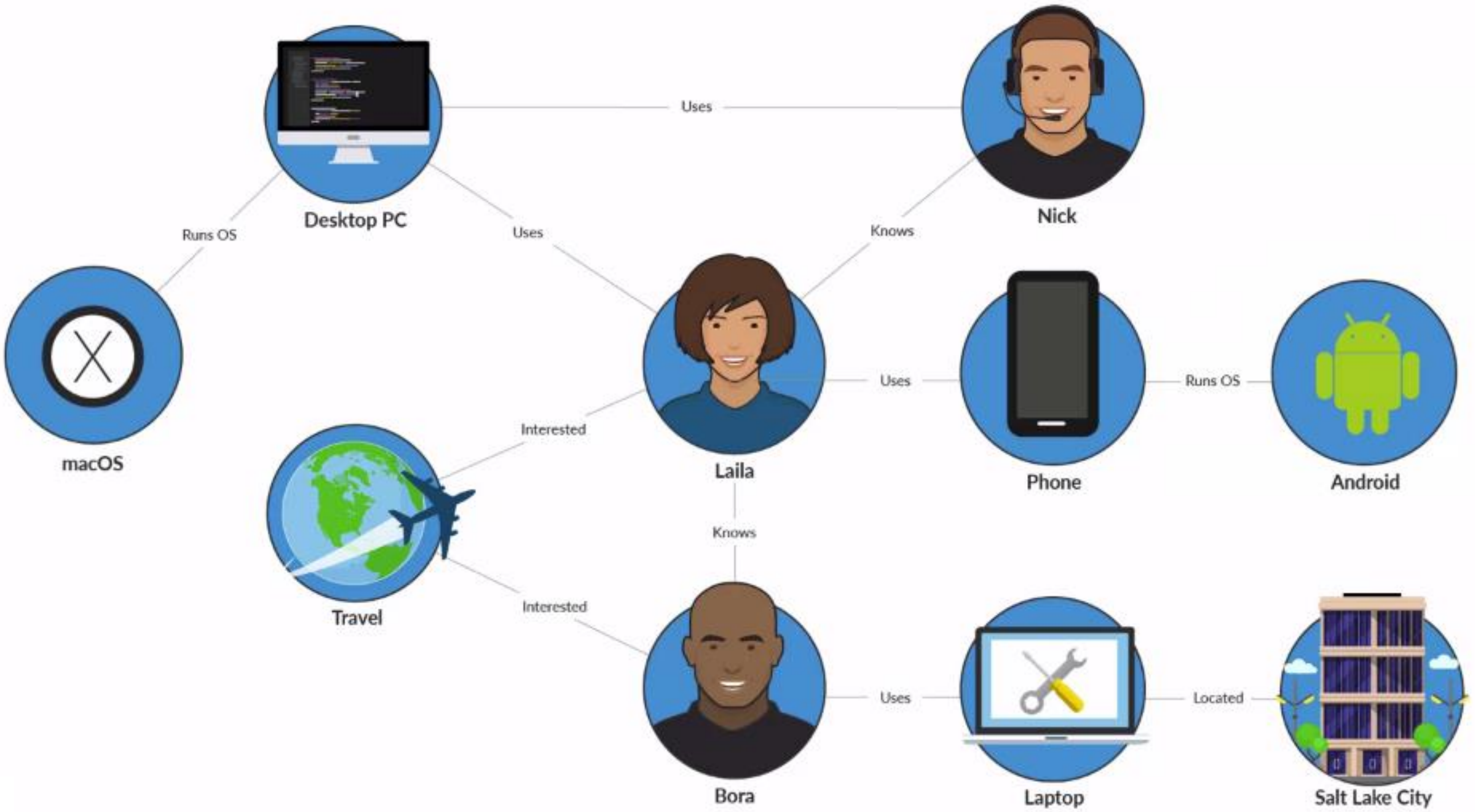
SQL API

SQL API

Lets you use a SQL-like language to query JSON documents (which is how Cosmos DB stores the data)

Easier for SQL users than the MongoDB API

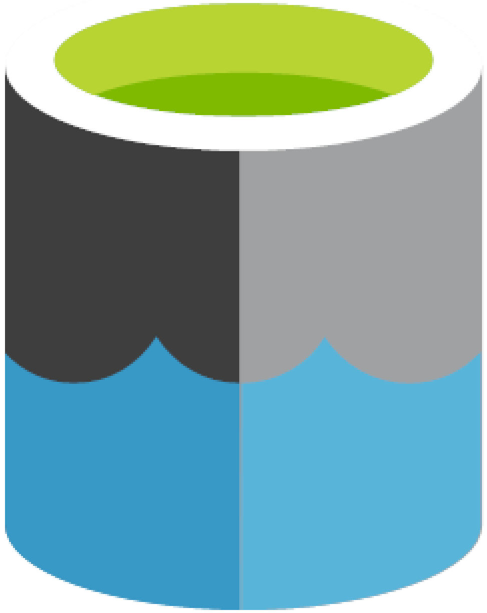
Graph Databases



Wide Column Model

- Used by Apache Cassandra
- Cosmos DB provides the Cassandra API for applications that are written to use a Cassandra database

Data Lake Storage Gen2



LEARNING OBJECTIVES

Get data into Azure Data Lake Storage (ADLS)

Use six layers of security to protect data in ADLS

Use Azure Databricks to process data in ADLS

Monitor and optimize the performance of your data lakes

Data Repositories

Data Warehouse

Structured data

Relational tables

Business reporting

Data Lake

Structured and raw data

Documents, images, social media, etc.

Data analytics and exploration

Unstructured Azure Data Services

Blob Storage

High availability

Lifecycle management

Low cost

Data Lake Storage Gen2

Blob Storage features plus:

Hierarchical

Fine-grained security

Hadoop compatibility

Azure Data Lake Storage (ADLS)

True hierarchical filesystem

- In Blob Storage, folders are just naming conventions
- ADLS is designed to perform operations on folders quickly

Fine-grained security

- Set permissions at individual file level

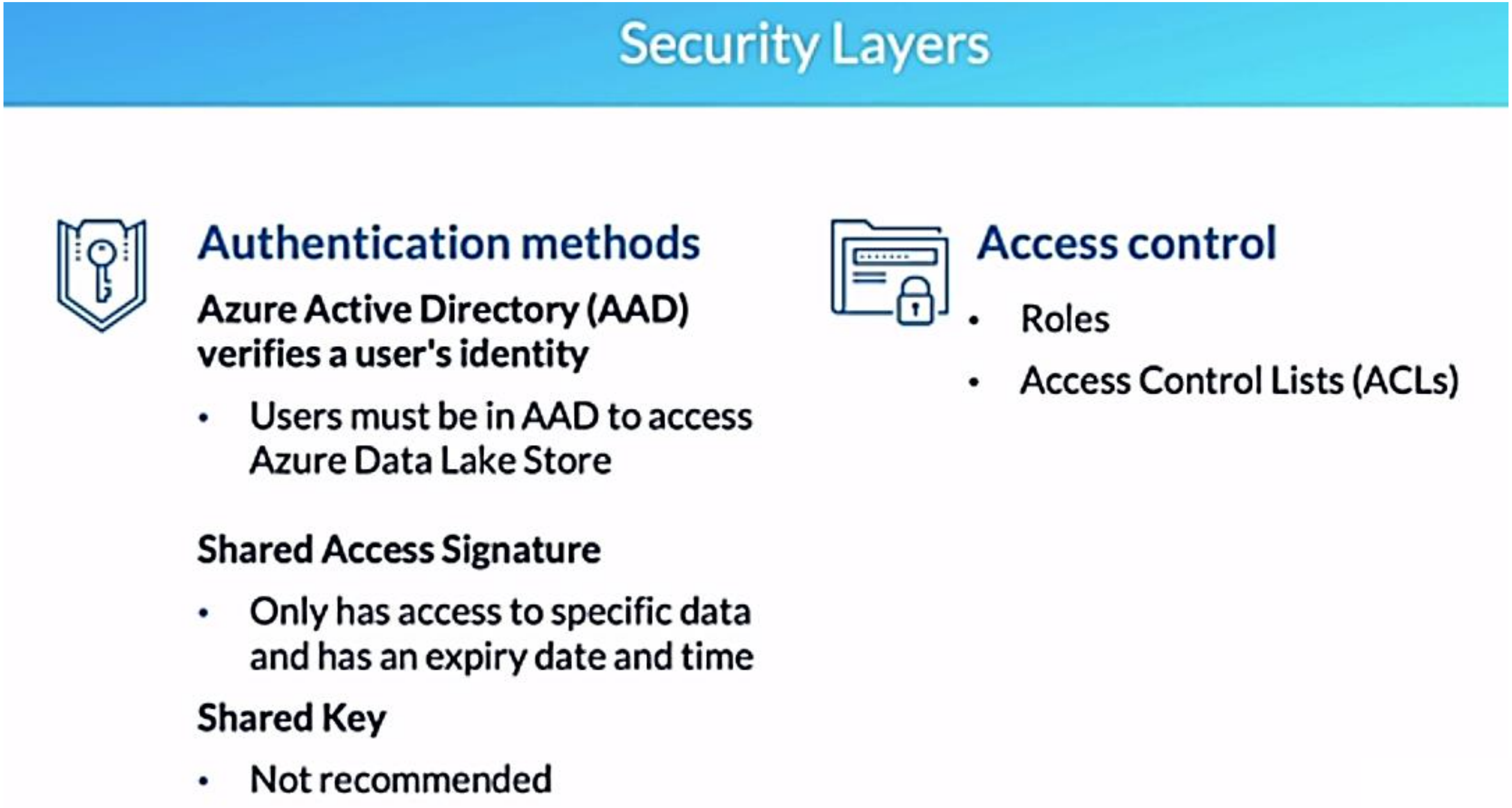
Can replace Hadoop Distributed File System (HDFS)

Use Spark on Azure Databricks to process data on ADLS

Security

Security layers

- Authentication
- Access control
- Network isolation
- Data protection
- Advanced threat protection
- Auditing



Security Layers

Authentication methods

Azure Active Directory (AAD) verifies a user's identity

- Users must be in AAD to access Azure Data Lake Store

Shared Access Signature

- Only has access to specific data and has an expiry date and time

Shared Key

- Not recommended

Access control

- Roles
- Access Control Lists (ACLs)

Roles

Owner

Read, write, and delete data

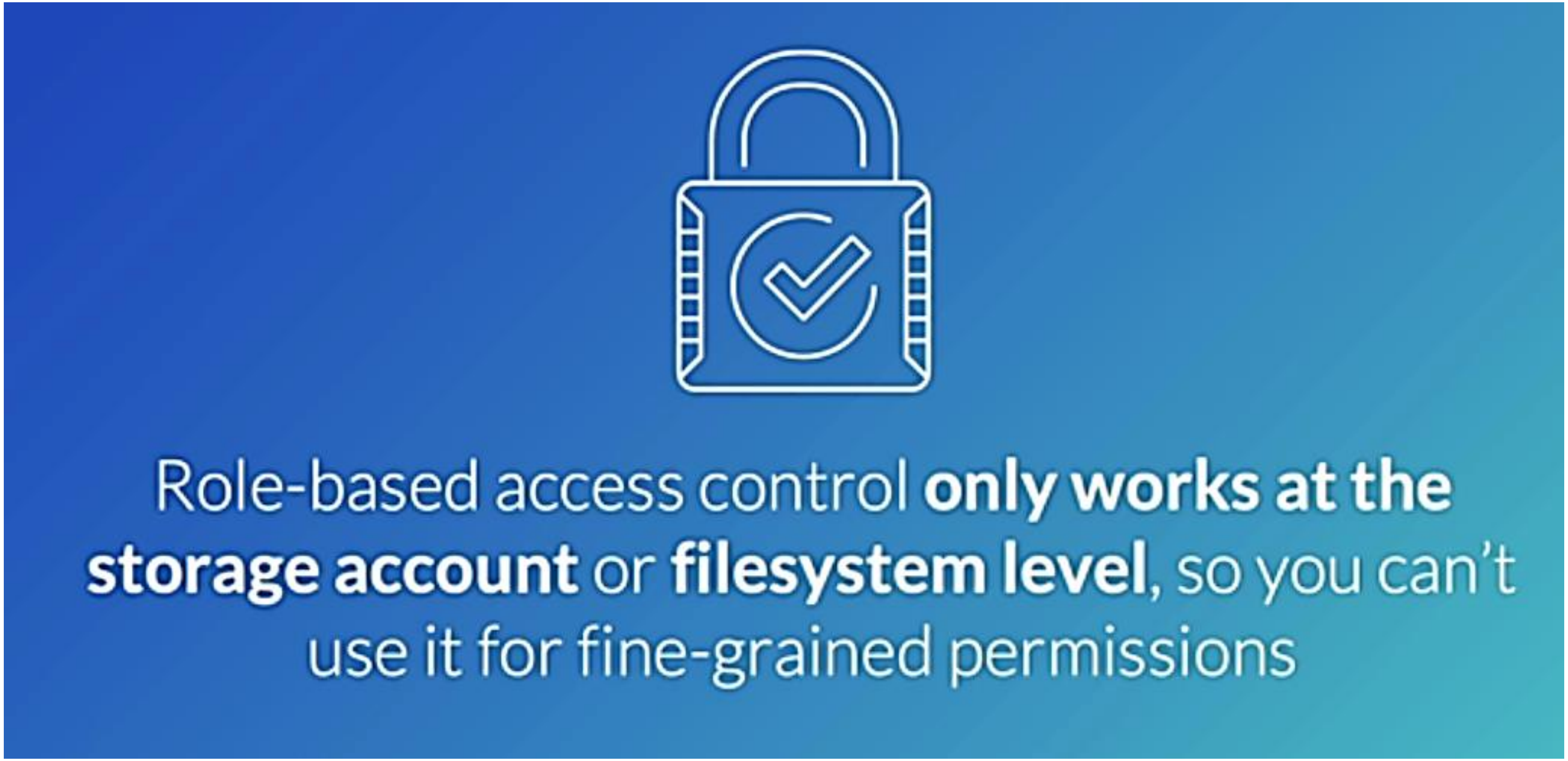
Set permissions of other users

Contributor

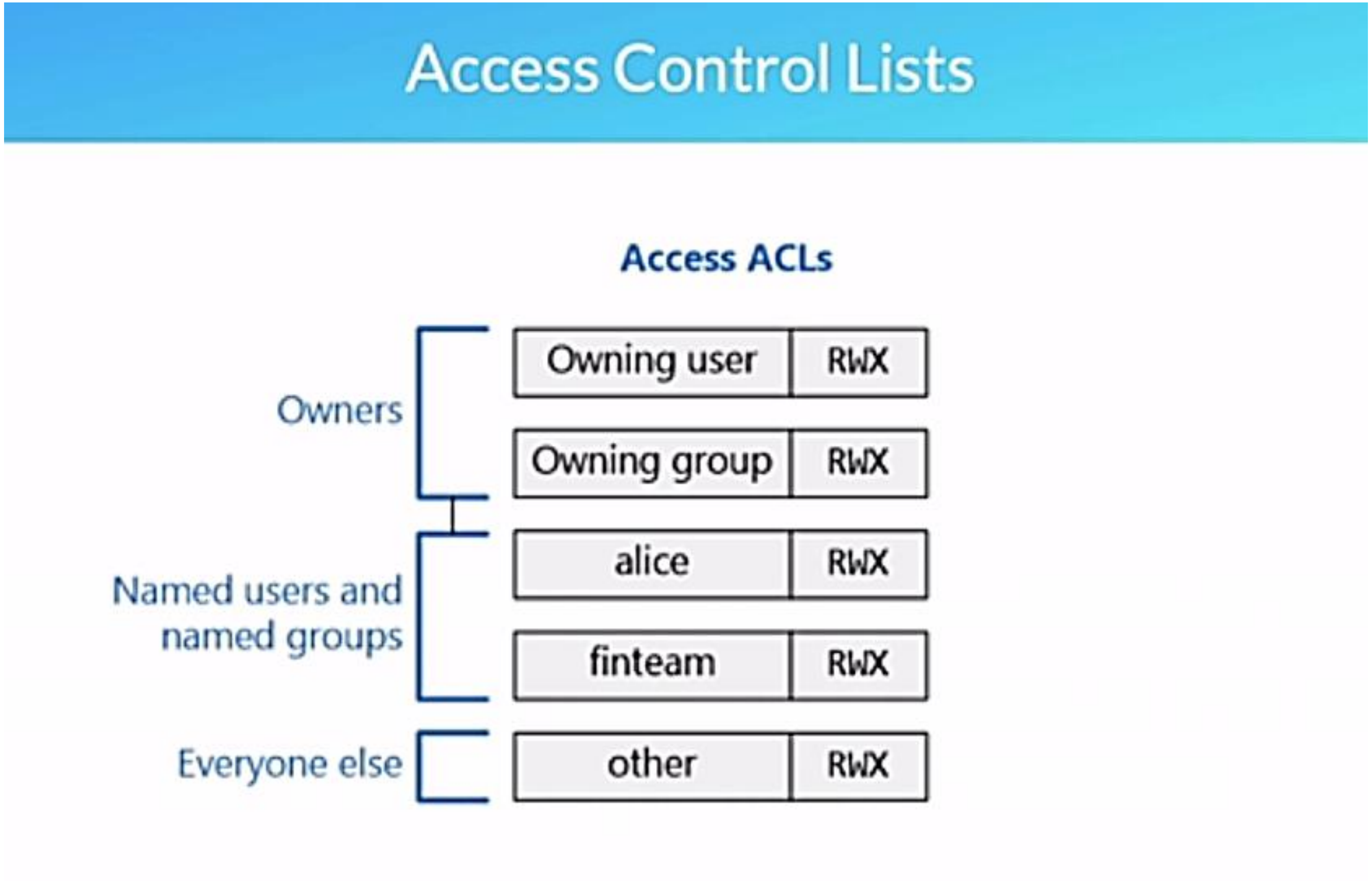
Read, write, and delete data

Reader

Read data



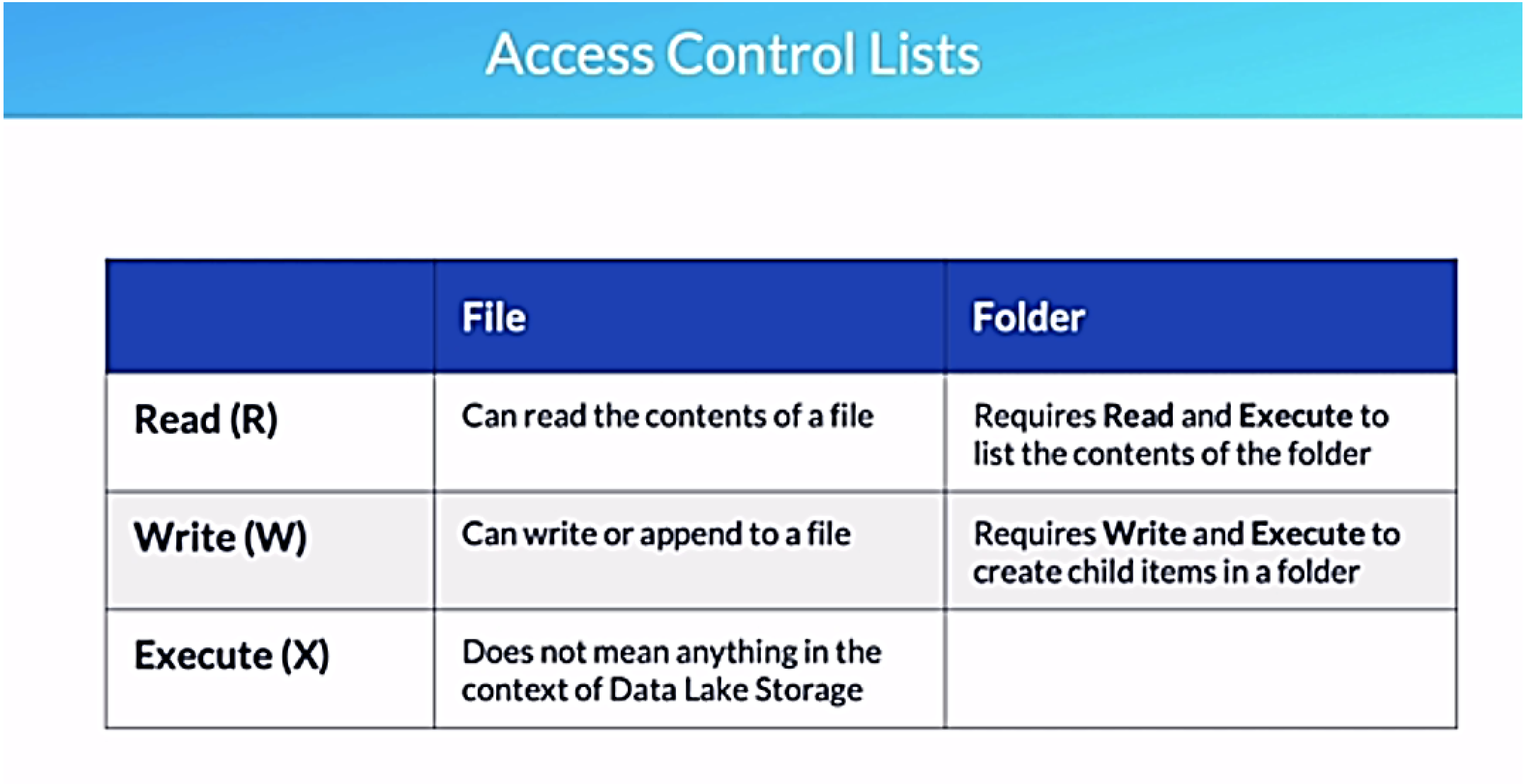
Role-based access control **only works at the storage account** or **filesystem level**, so you can’t use it for fine-grained permissions



Access Control Lists

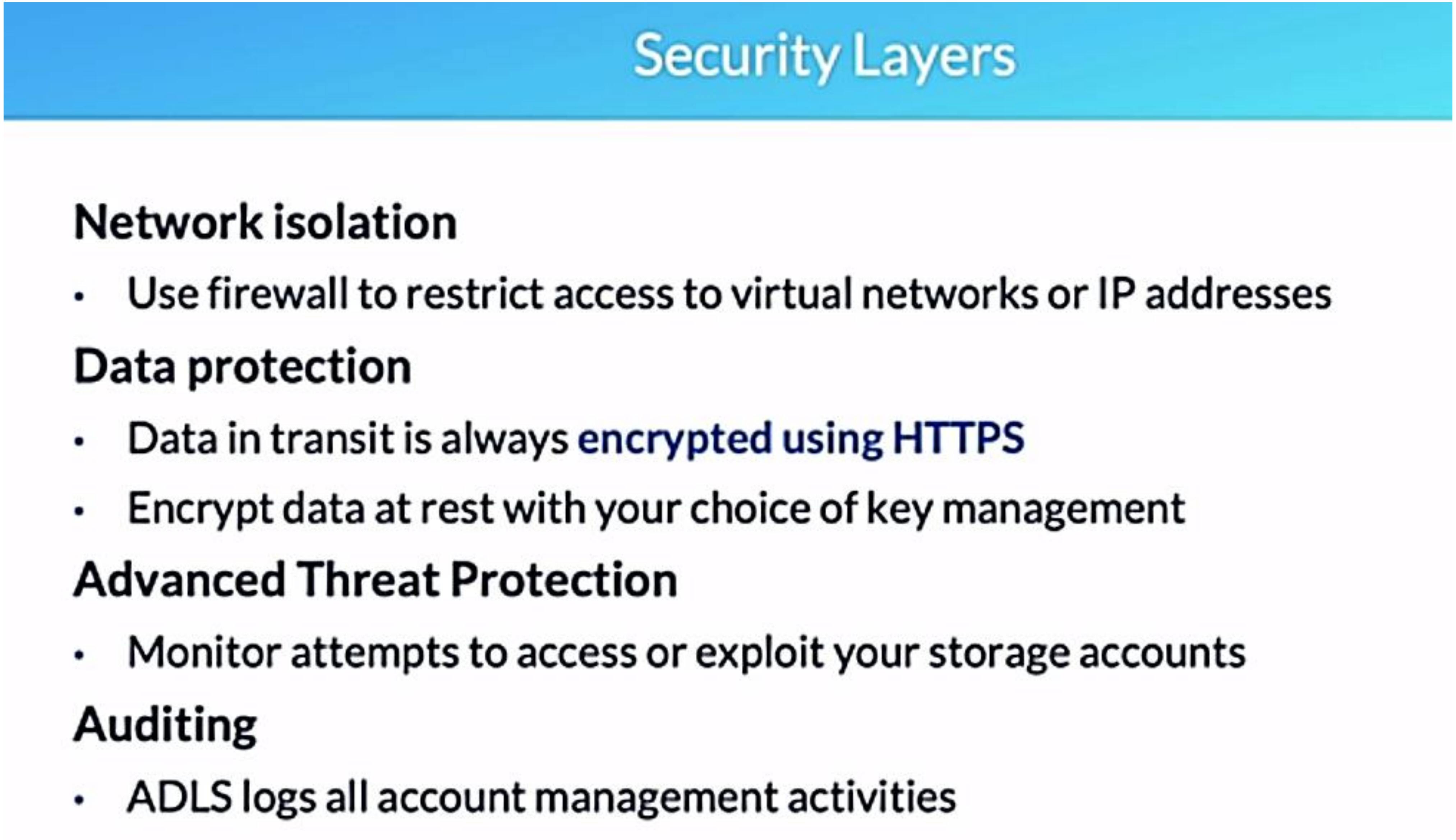
Access ACLs

Owners	Owning user	RWX
	Owning group	RWX
Named users and named groups	alice	RWX
	finteam	RWX
Everyone else	other	RWX



Access Control Lists

	File	Folder
Read (R)	Can read the contents of a file	Requires **Read** and **Execute** to list the contents of the folder
Write (W)	Can write or append to a file	Requires **Write** and **Execute** to create child items in a folder
Execute (X)	Does not mean anything in the context of Data Lake Storage	



Security Layers

Network isolation

- Use firewall to restrict access to virtual networks or IP addresses

Data protection

- Data in transit is always **encrypted using HTTPS**
- Encrypt data at rest with your choice of key management

Advanced Threat Protection

- Monitor attempts to access or exploit your storage accounts

Auditing

- ADLS logs all account management activities

Ingesting Data

Ways to upload data from your desktop to ADLS

- AzCopy
- Azure Storage Explorer
- PowerShell
- Azure CLI

I'll use AzCopy from Cloud Shell

Need to assign Storage Blob Data Contributor role first

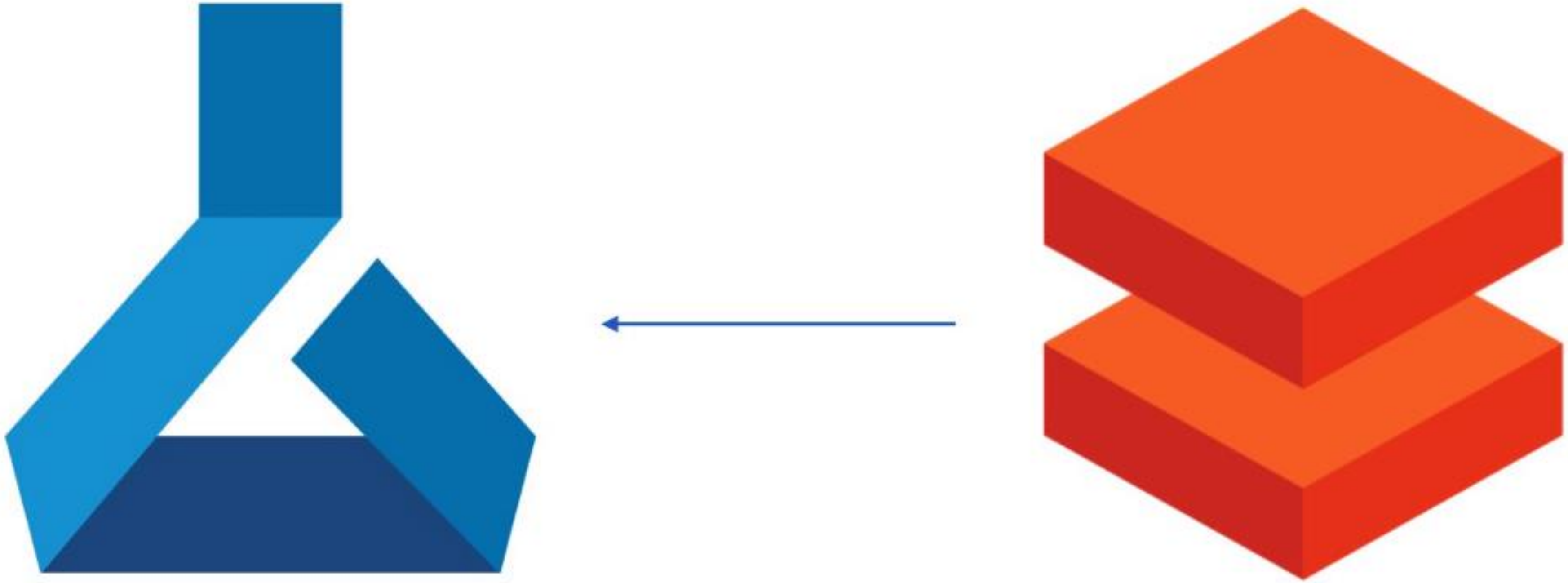
Accessing ADLS from Azure Databricks

Overview

- In 2013, the creators of Spark started a company called Databricks
- The name of their product is also Databricks
- It's a managed implementation of Spark in the cloud
- It also has a user-friendly interface for running code on clusters interactively

Databricks

Overview



Azure Machine Learning Services

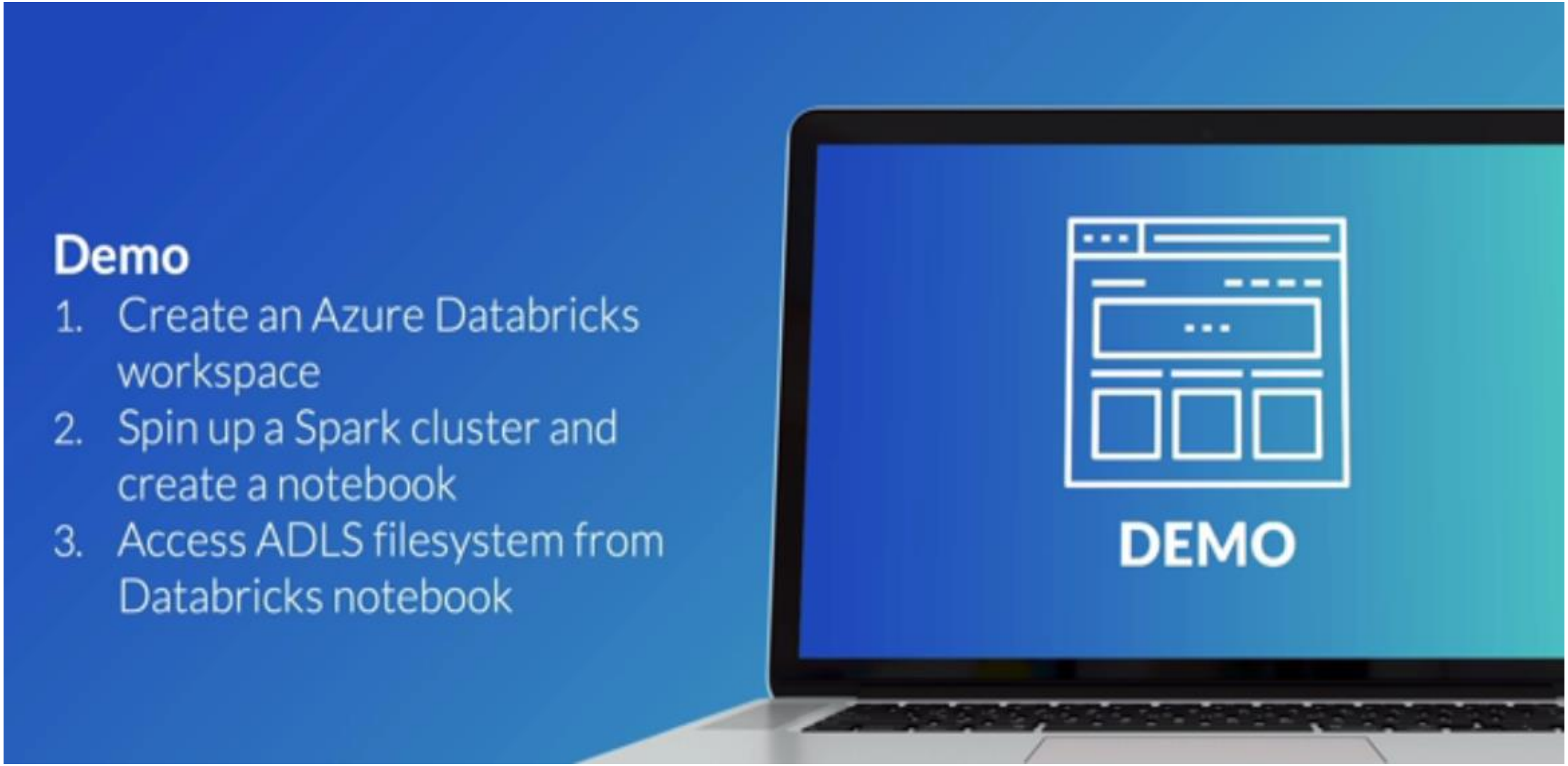
Databricks

Accessing ADLS from Azure Databricks

Ways to authenticate Databricks to ADLS

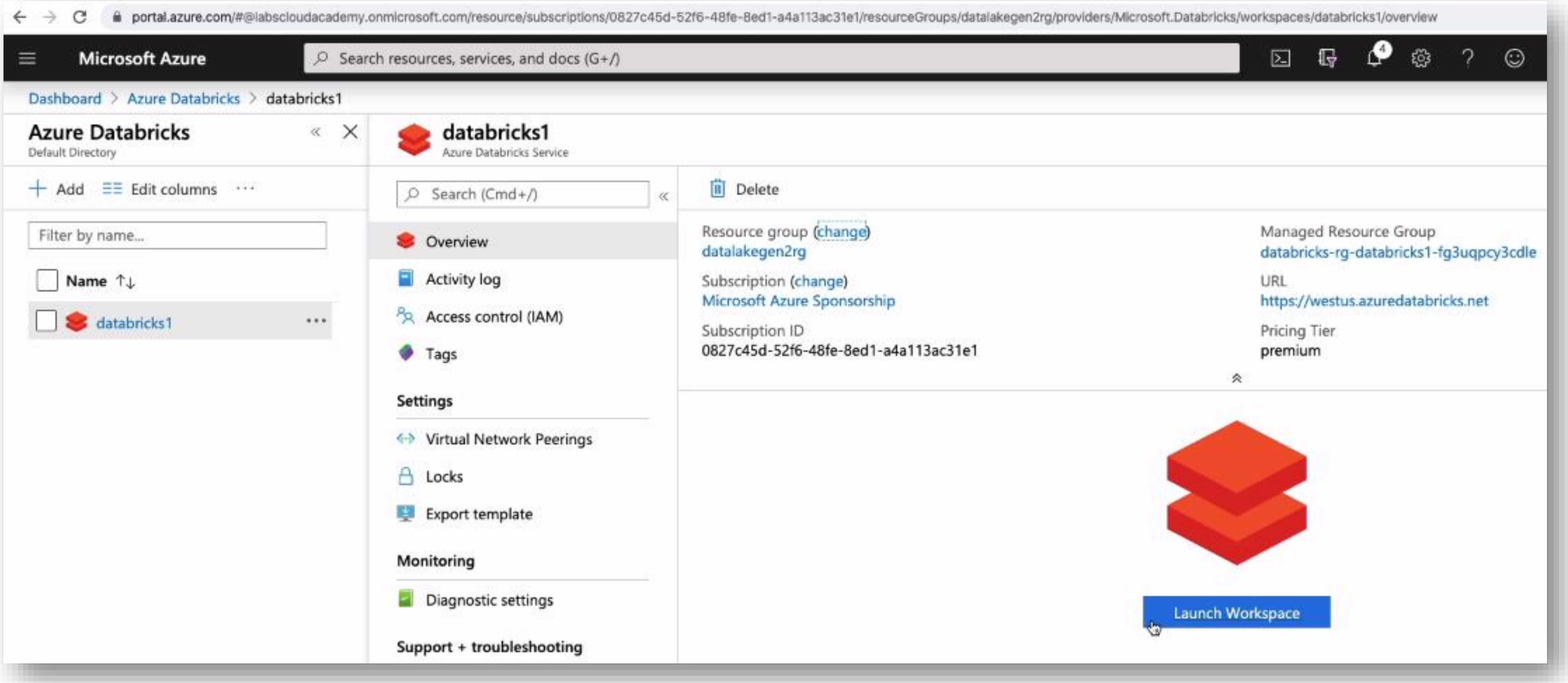
- Credential passthrough - uses Azure AD credentials
- Service principal - identity you assign to a service
- Embed storage account access key in code on Databricks – not recommended

Credential passthrough only works on Premium Databricks workspace



Demo

1. Create an Azure Databricks workspace
2. Spin up a Spark cluster and create a notebook
3. Access ADLS filesystem from Databricks notebook



portal.azure.com/#@labscloudacademy.onmicrosoft.com/resource/subscriptions/0827c45d-52f6-48fe-8ed1-a4a113ac31e1/resourceGroups/datalakegen2rg/providers/Microsoft.Databricks/workspaces/databricks1/overview

Dashboard > Azure Databricks > databricks1

Azure Databricks
Default Directory

□ Name ↑↓
□ databricks1

databricks1
Azure Databricks Service

Resource group (change)
datalakegen2rg

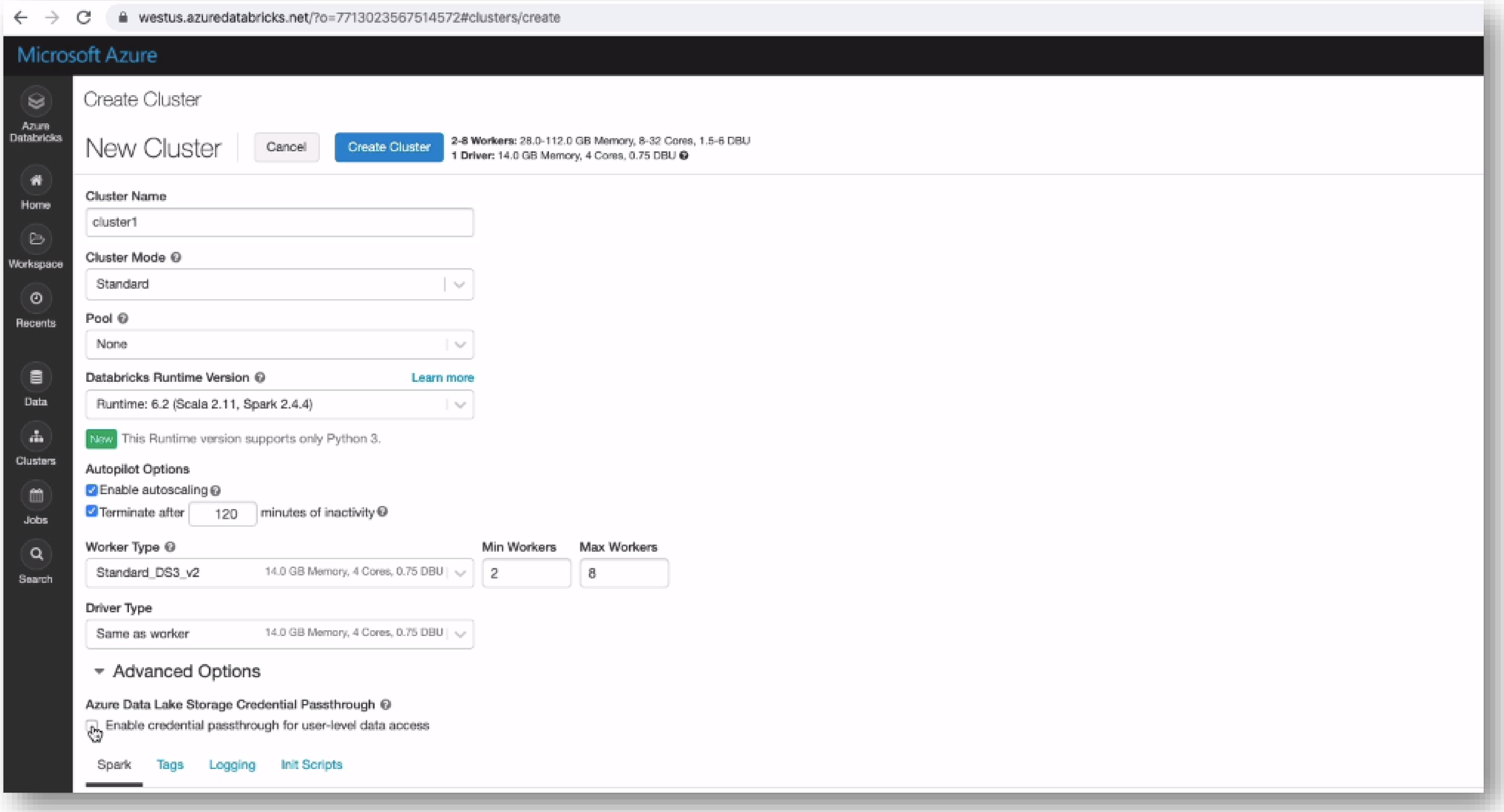
Subscription (change)
Microsoft Azure Sponsorship

Subscription ID
0827c45d-52f6-48fe-8ed1-a4a113ac31e1

Managed Resource Group
databricks-rg-databricks1-fg3uqpcy3cdle

URL
https://westus.azuredatabricks.net

Pricing Tier
premium



Create Cluster

New Cluster

2-8 Workers: 28.0-112.0 GB Memory, 8-32 Cores, 1.5-6 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: cluster1

Cluster Mode: Standard

Pool: None

Databricks Runtime Version: Runtime: 6.2 (Scala 2.11, Spark 2.4.4)

New This Runtime version supports only Python 3.

Autopilot Options

☑ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

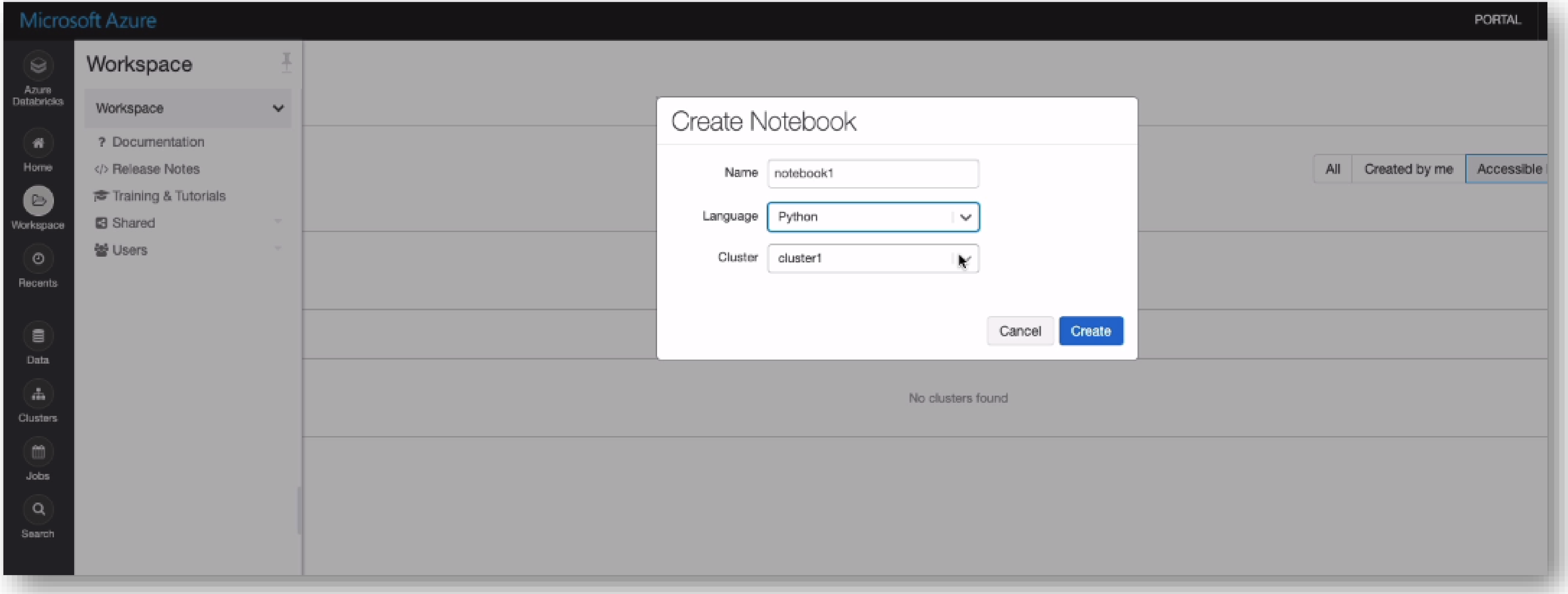
Worker Type	Min Workers	Max Workers
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU	2	8

Driver Type: Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU

Advanced Options

Azure Data Lake Storage Credential Passthrough

☐ Enable credential passthrough for user-level data access



Create Notebook

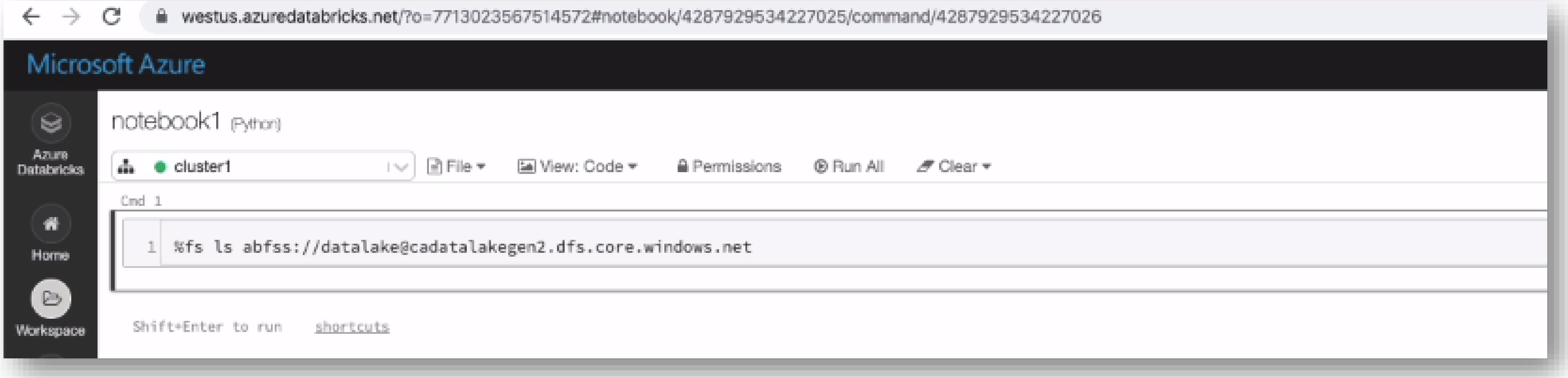
Name: notebook1

Language: Python

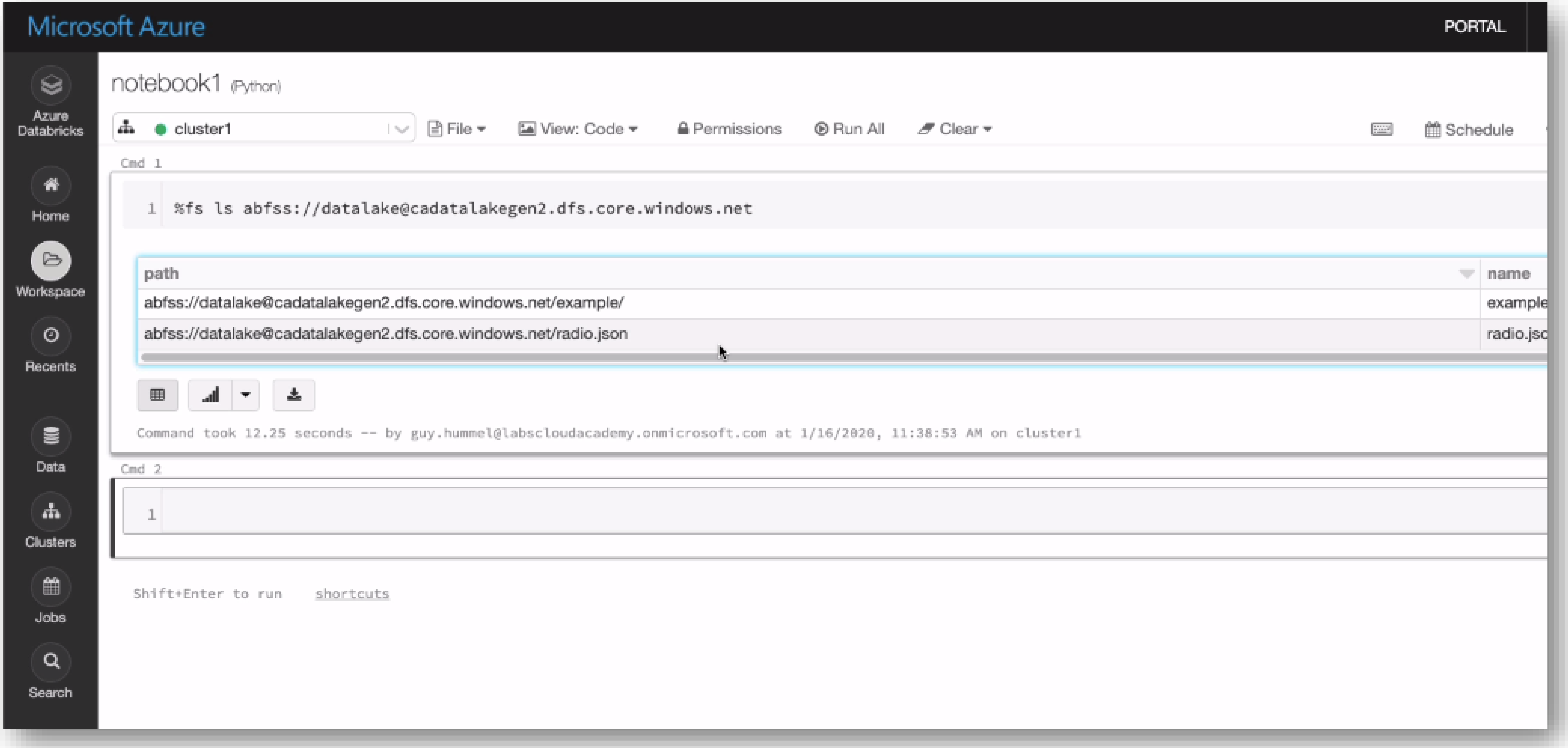
Cluster: cluster1

Cancel | Create

No clusters found



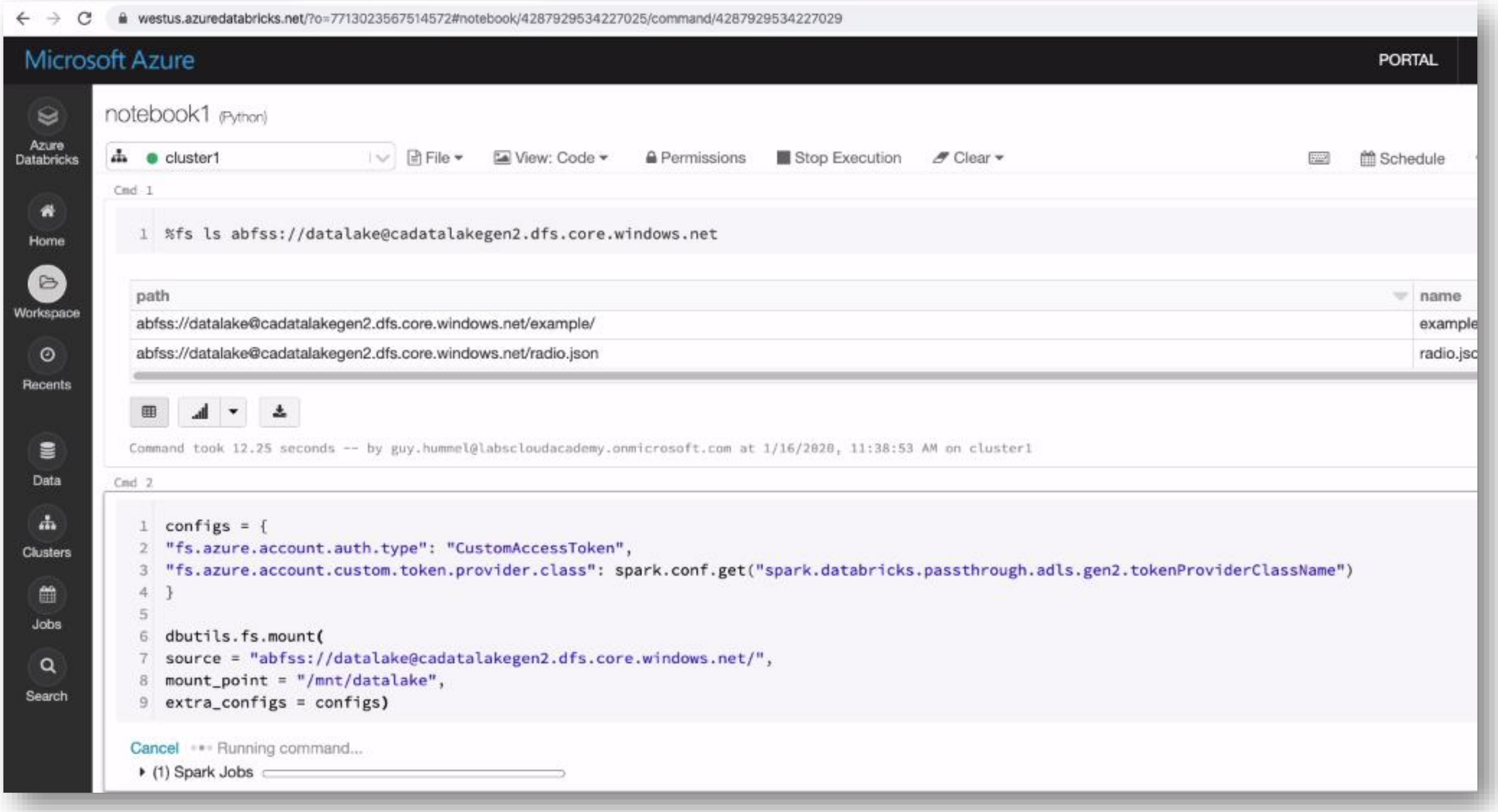
```
%fs ls abfss://datalake@cadatalakegen2.dfs.core.windows.net
```



```
%fs ls abfss://datalake@cadatalakegen2.dfs.core.windows.net
```

path	name
abfss://datalake@cadatalakegen2.dfs.core.windows.net/example/	example
abfss://datalake@cadatalakegen2.dfs.core.windows.net/radio.json	radio.jso

```
Command took 12.25 seconds -- by guy.hummel@labscloudacademy.onmicrosoft.com at 1/16/2020, 11:38:53 AM on cluster1
```

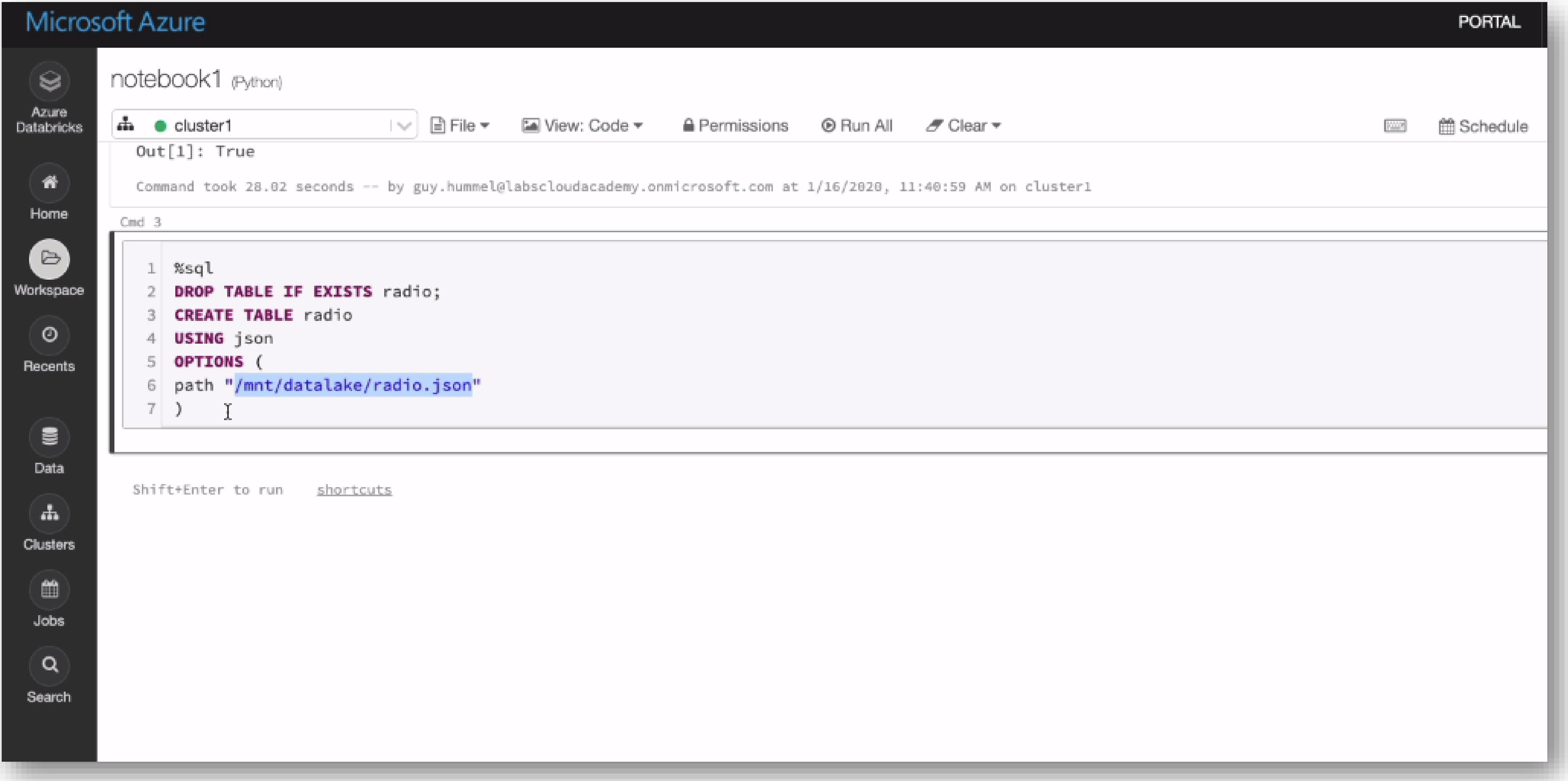


```
%fs ls abfss://datalake@cadatalakegen2.dfs.core.windows.net
```

path	name
abfss://datalake@cadatalakegen2.dfs.core.windows.net/example/	example
abfss://datalake@cadatalakegen2.dfs.core.windows.net/radio.json	radio.jsc

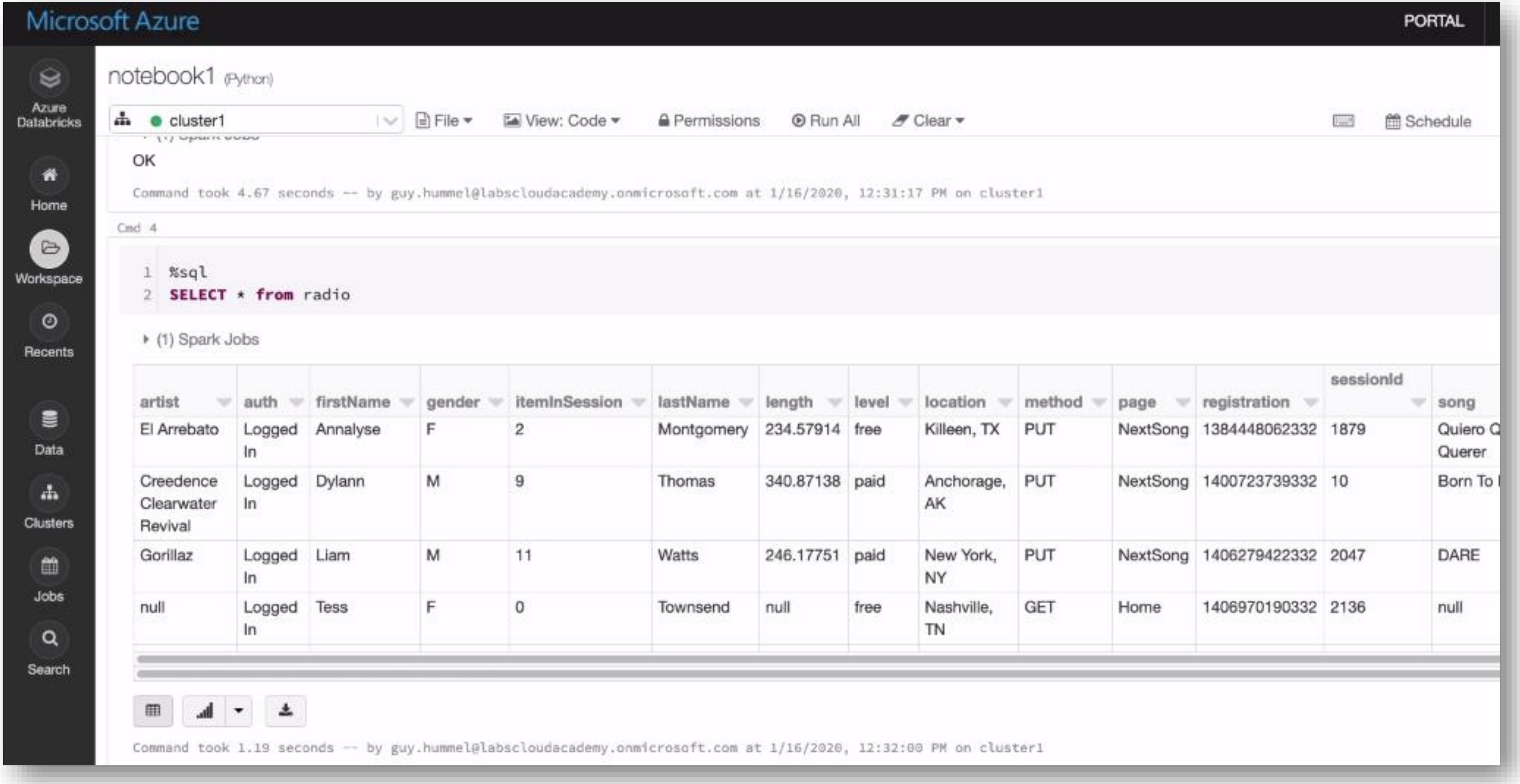
```
Command took 12.25 seconds -- by guy.hummel@labscloudacademy.onmicrosoft.com at 1/16/2020, 11:38:53 AM on cluster1
```

```
configs = {
"fs.azure.account.auth.type": "CustomAccessToken",
"fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")
}

dbutils.fs.mount(
source = "abfss://datalake@cadatalakegen2.dfs.core.windows.net/",
mount_point = "/mnt/datalake",
extra_configs = configs)
```



```
Out[1]: True

Command took 28.02 seconds -- by guy.hummel@labscloudacademy.onmicrosoft.com at 1/16/2020, 11:40:59 AM on cluster1
```

```
%sql
DROP TABLE IF EXISTS radio;
CREATE TABLE radio
USING json
OPTIONS (
path "/mnt/datalake/radio.json"
)
```

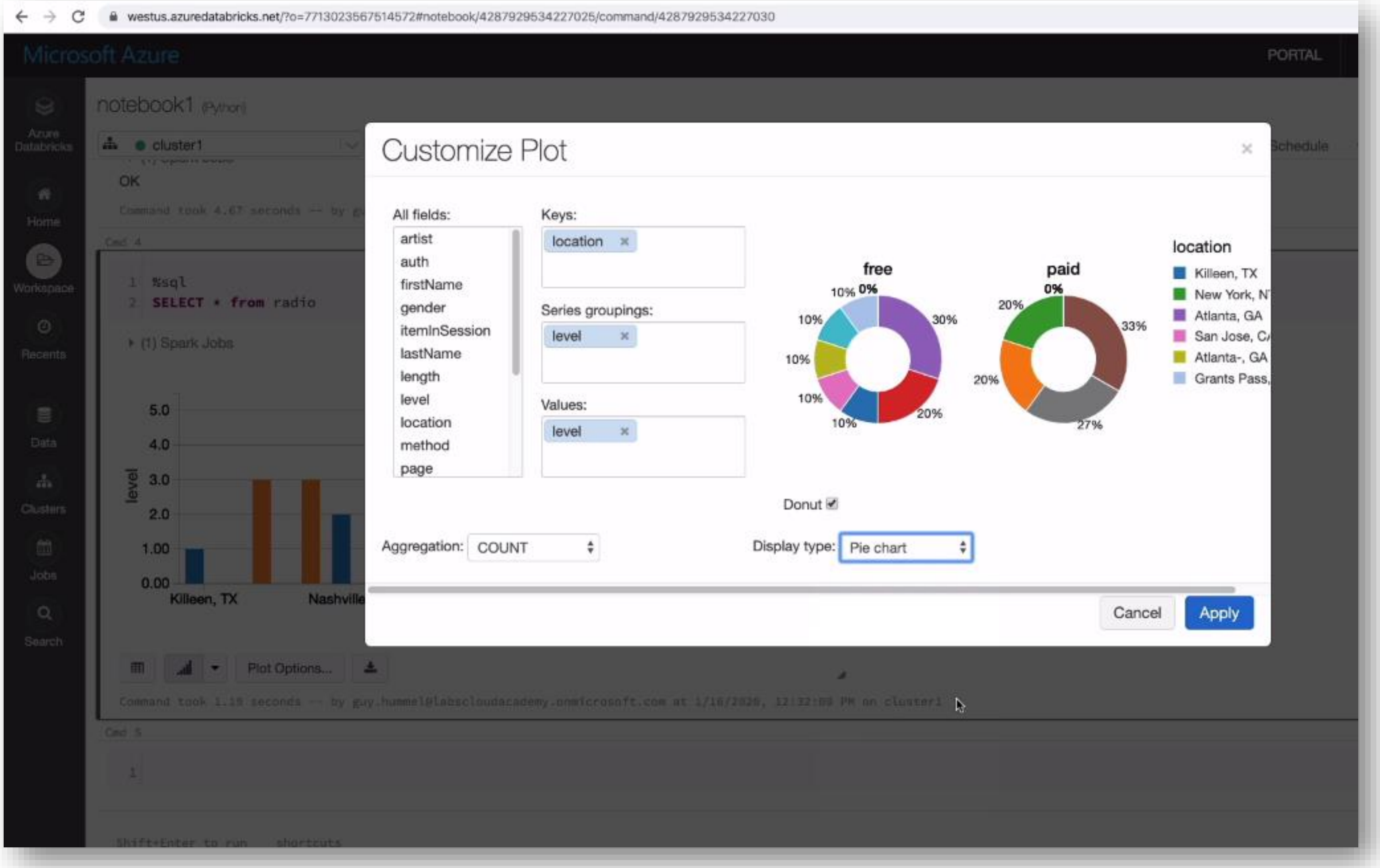


```
OK
Command took 4.67 seconds -- by guy.hummel@labscloudacademy.onmicrosoft.com at 1/16/2020, 12:31:17 PM on cluster1
```

```
%sql
SELECT * from radio
```

artist	auth	firstName	gender	itemInSession	lastName	length	level	location	method	page	registration	sessionId	song
El Arrebato	Logged In	Annalyse	F	2	Montgomery	234.57914	free	Killeen, TX	PUT	NextSong	1384448062332	1879	Quiero Q Querer
Creedence Clearwater Revival	Logged In	Dylann	M	9	Thomas	340.87138	paid	Anchorage, AK	PUT	NextSong	1400723739332	10	Born To I
Gorillaz	Logged In	Liam	M	11	Watts	246.17751	paid	New York, NY	PUT	NextSong	1406279422332	2047	DARE
null	Logged In	Tess	F	0	Townsend	null	free	Nashville, TN	GET	Home	1406970190332	2136	null

```
Command took 1.19 seconds -- by guy.hummel@labscloudacademy.onmicrosoft.com at 1/16/2020, 12:32:00 PM on cluster1
```



westus.azuredatabricks.net/?o=7713023567514572#notebook/4287929534227025/command/4287929534227030

notebook1 (Python)

cluster1

OK

```
Command took 4.67 seconds -- by g
```

```
%sql
SELECT * from radio
```

(1) Spark Jobs

```
Command took 1.18 seconds -- by guy.hummel@labscloudacademy.onmicrosoft.com at 1/16/2020, 12:32:09 PM on cluster1
```

Shift+Enter to run   shortcuts

Customize Plot

All fields: artist, auth, firstName, gender, itemInSession, lastName, length, level, location, method, page

Keys: location

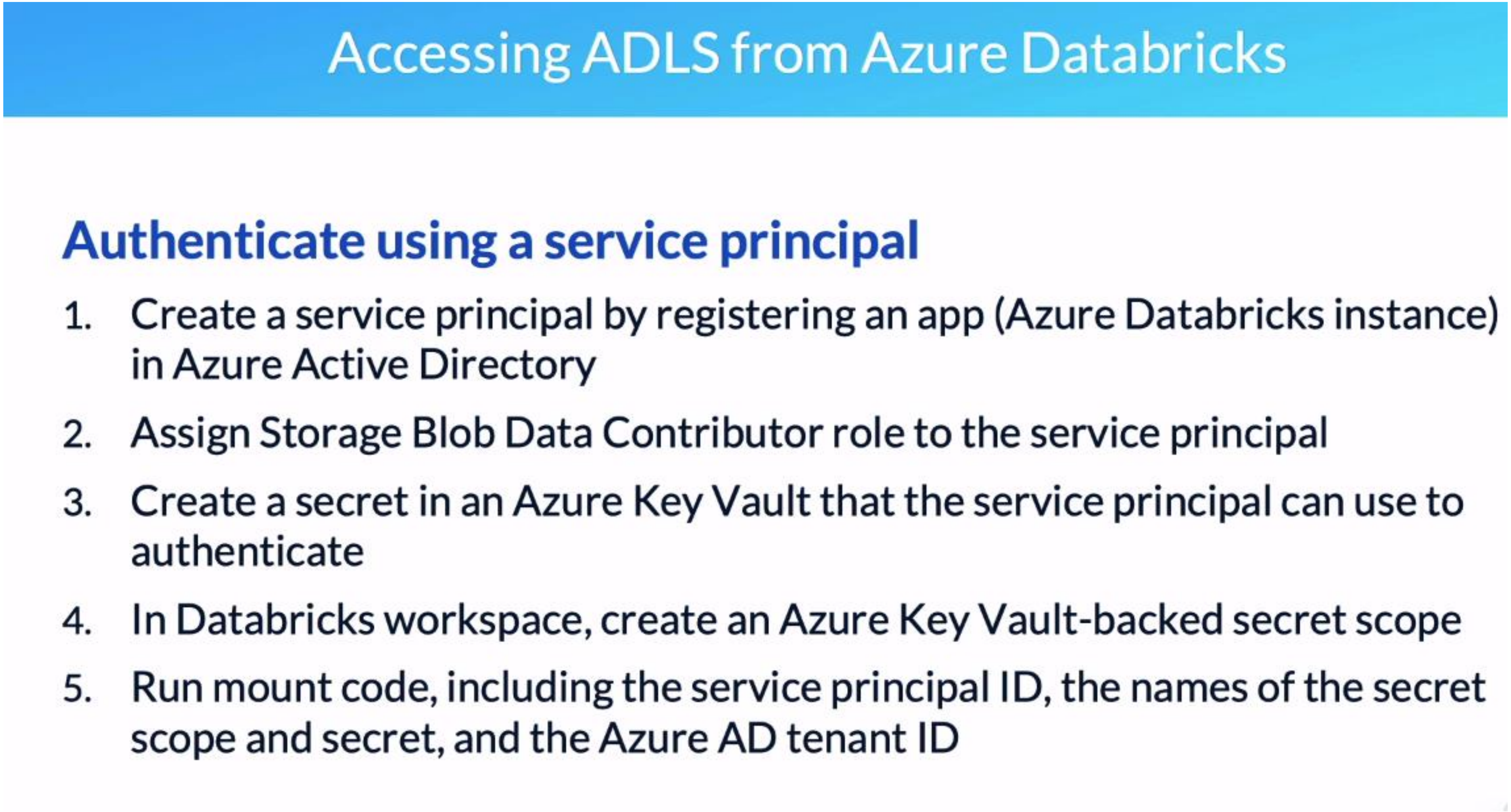
Series groupings: level

Values: level

Donut ☑

Aggregation: COUNT

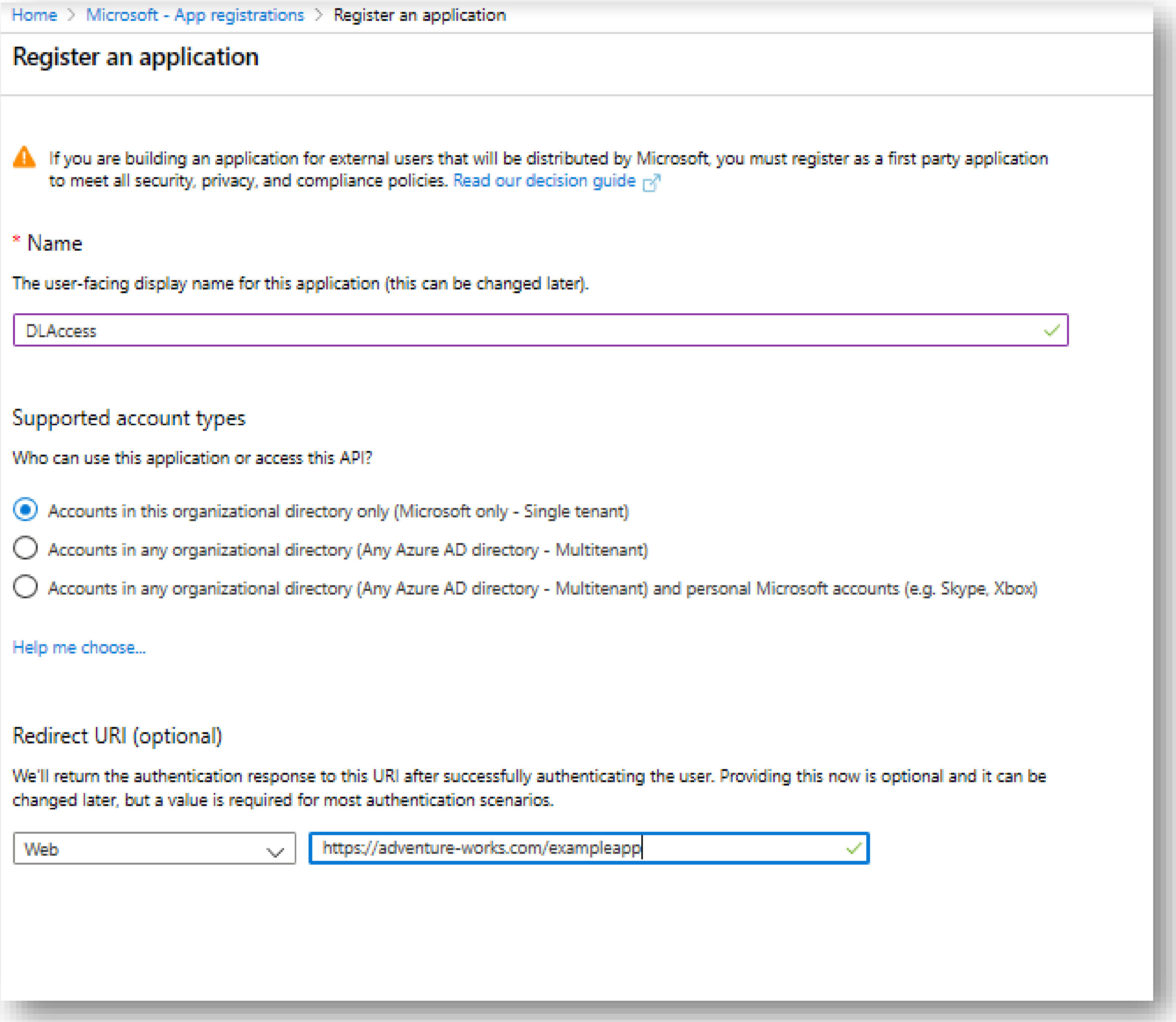
Display type: Pie chart



Accessing ADLS from Azure Databricks

Authenticate using a service principal

1. Create a service principal by registering an app (Azure Databricks instance) in Azure Active Directory
2. Assign Storage Blob Data Contributor role to the service principal
3. Create a secret in an Azure Key Vault that the service principal can use to authenticate
4. In Databricks workspace, create an Azure Key Vault-backed secret scope
5. Run mount code, including the service principal ID, the names of the secret scope and secret, and the Azure AD tenant ID



Home > Microsoft - App registrations > Register an application

Register an application

⚠ If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. Read our decision guide

* Name

The user-facing display name for this application (this can be changed later).

DLAccess ✓

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Microsoft only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

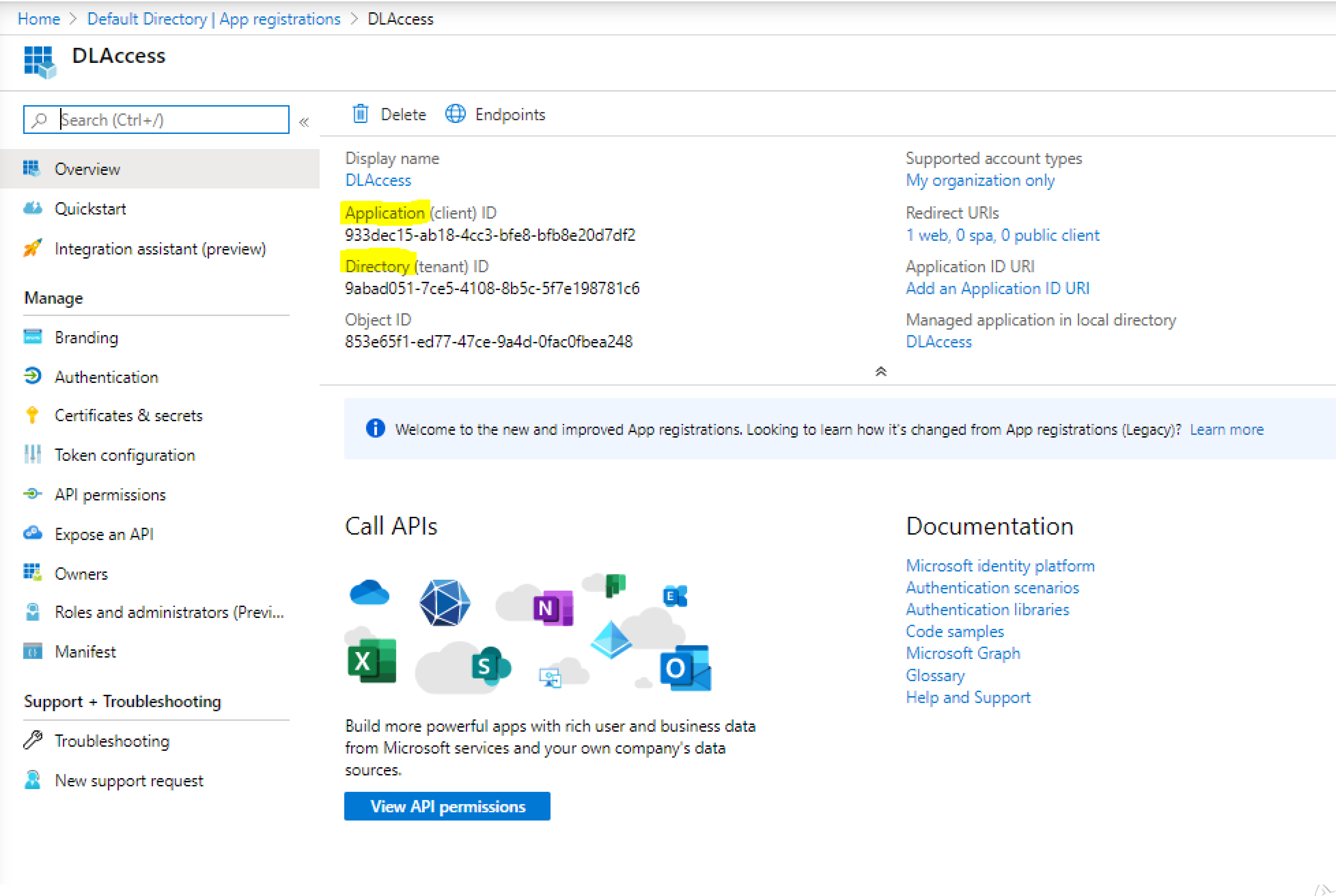
Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web

https://adventure-works.com/exampleapp ✓



DLAccess

Display name
DLAccess

Application (client) ID
933dec15-ab18-4cc3-bfe8-bfb8e20d7df2

Directory (tenant) ID
9abad051-7ce5-4108-8b5c-5f7e198781c6

Object ID
853e65f1-ed77-47ce-9a4d-0fac0fbea248

Supported account types
My organization only

Redirect URIs
1 web, 0 spa, 0 public client

Application ID URI
Add an Application ID URI

Managed application in local directory
DLAccess

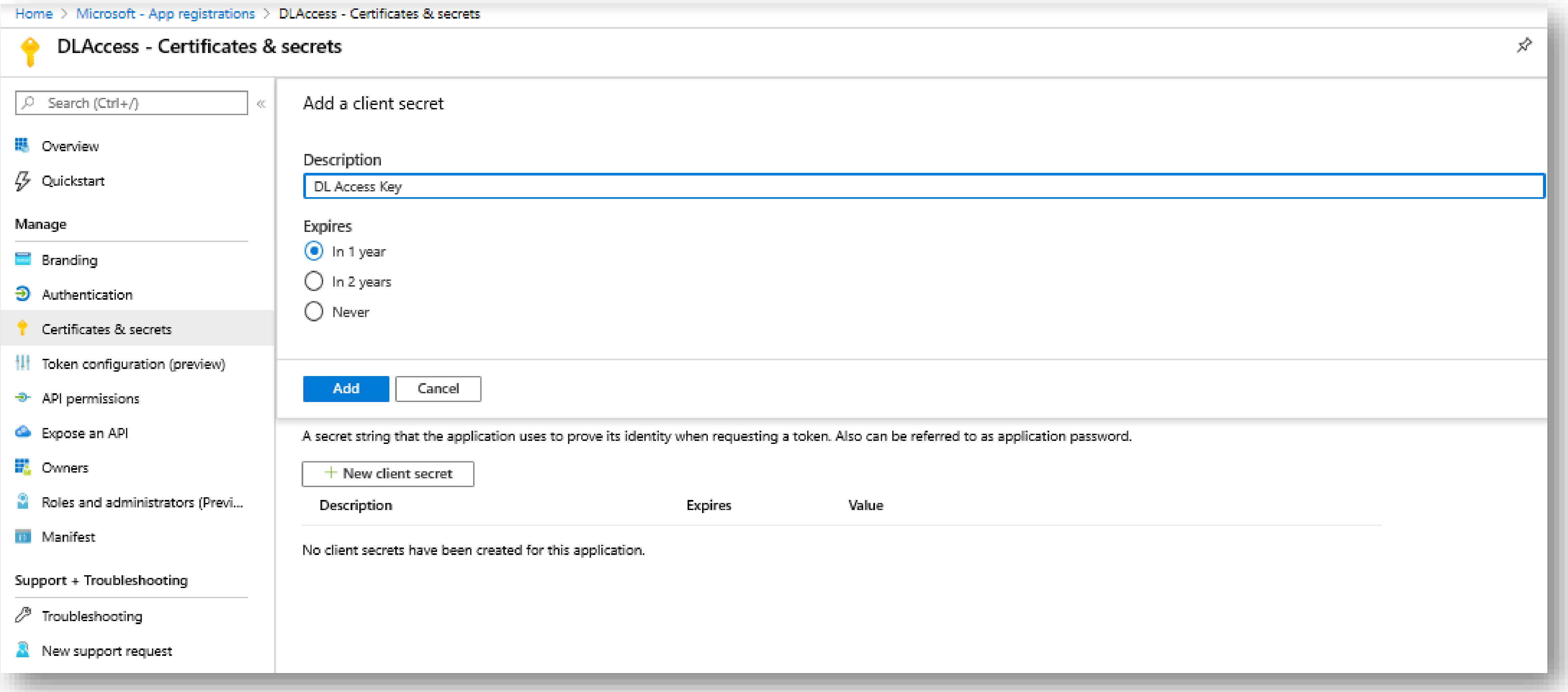
Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

Documentation

Microsoft identity platform
Authentication scenarios
Authentication libraries
Code samples
Microsoft Graph
Glossary
Help and Support


Home > Microsoft - App registrations > DLAccess - Certificates & secrets
DLAccess - Certificates & secrets
Search (Ctrl+/)
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
Token configuration (preview)
API permissions
Expose an API
Owners
Roles and administrators (Previ...
Manifest
Support + Troubleshooting
Troubleshooting
New support request

Add a client secret

Description

DL Access Key

Expires

● In 1 year

○ In 2 years

○ Never

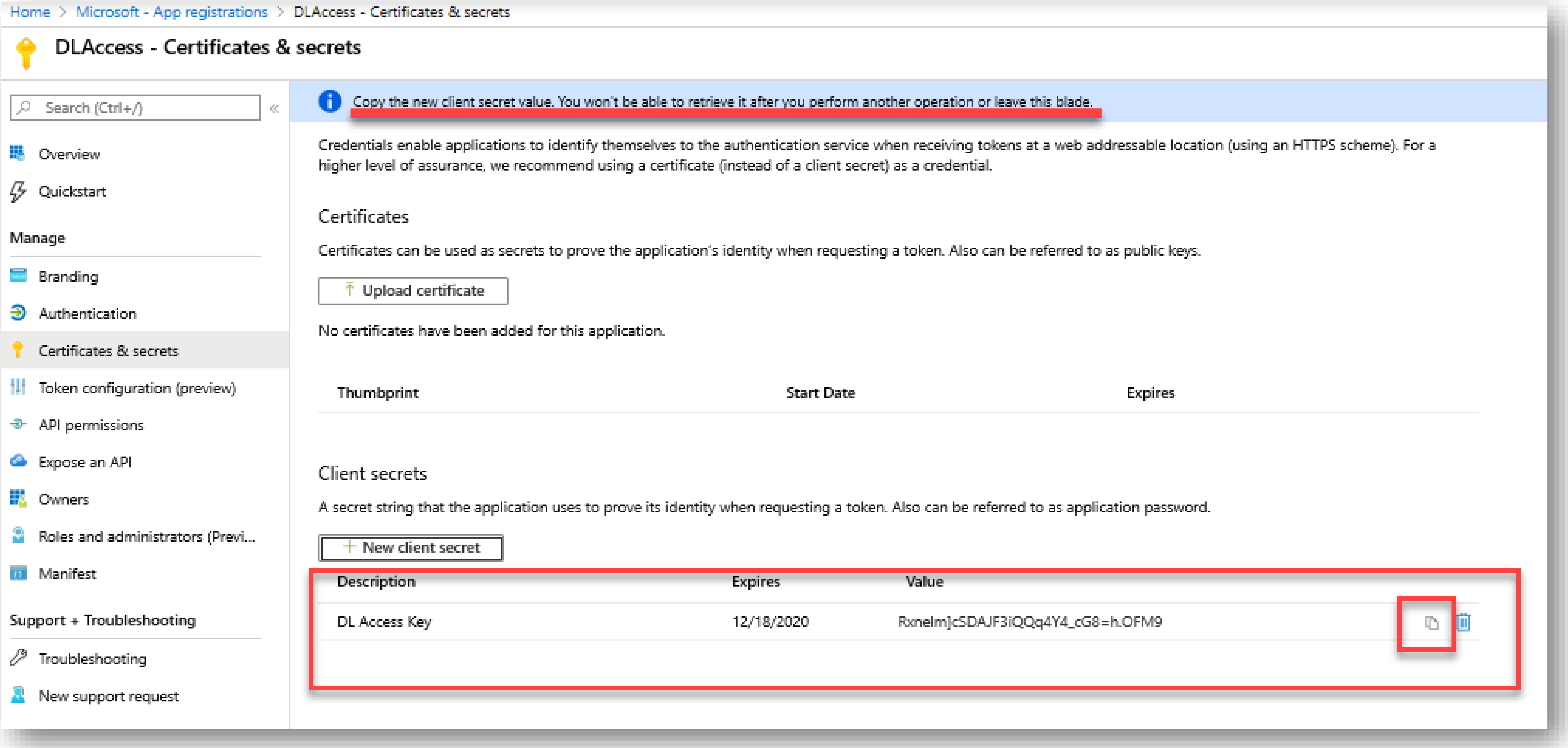
Add   Cancel

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value

No client secrets have been created for this application.



DLAccess - Certificates & secrets

Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

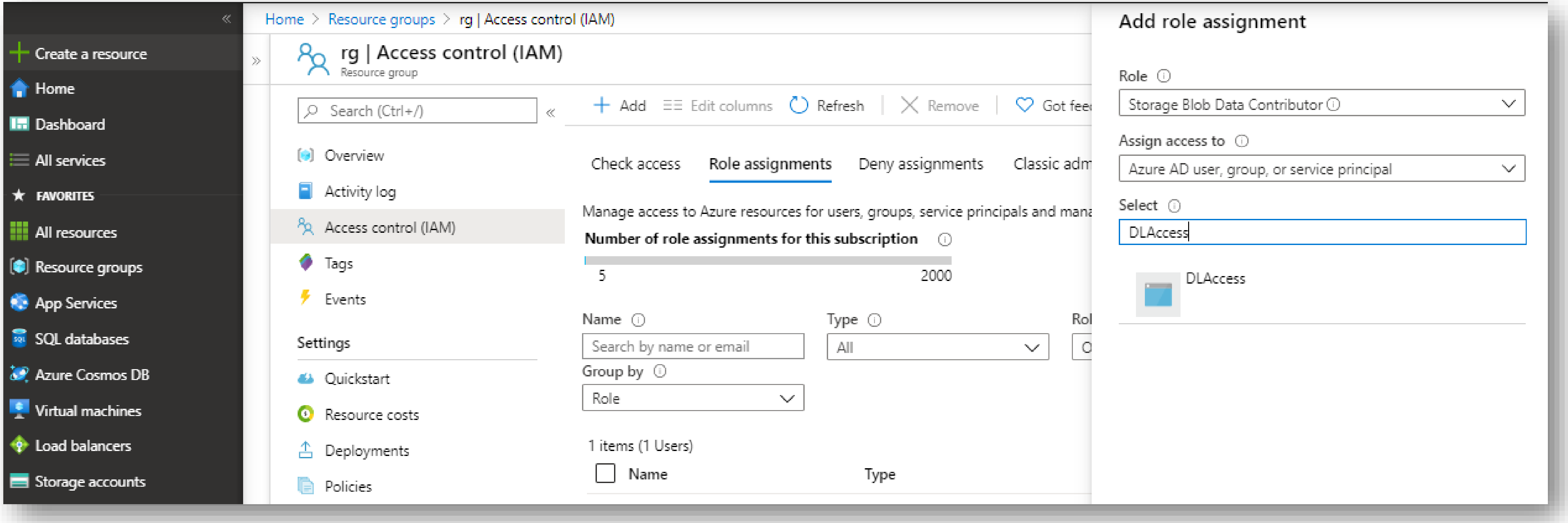
No certificates have been added for this application.

Thumbprint	Start Date	Expires

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value
DL Access Key	12/18/2020	Rxnelm]cSDAJF3iQQq4Y4_cG8=h.OFM9



Manage access to Azure resources for users, groups, service principals and mana

Number of role assignments for this subscription

5 2000

Name: Search by name or email

Type: All

Rol: O

Group by: Role

1 items (1 Users)

☐ Name	Type

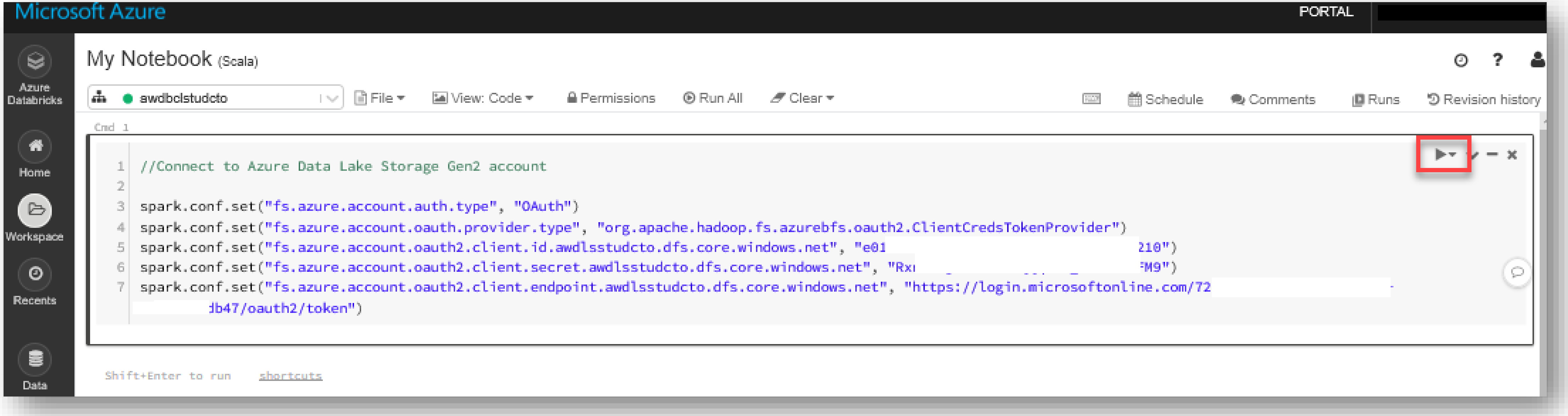
Add role assignment

Role: Storage Blob Data Contributor

Assign access to: Azure AD user, group, or service principal

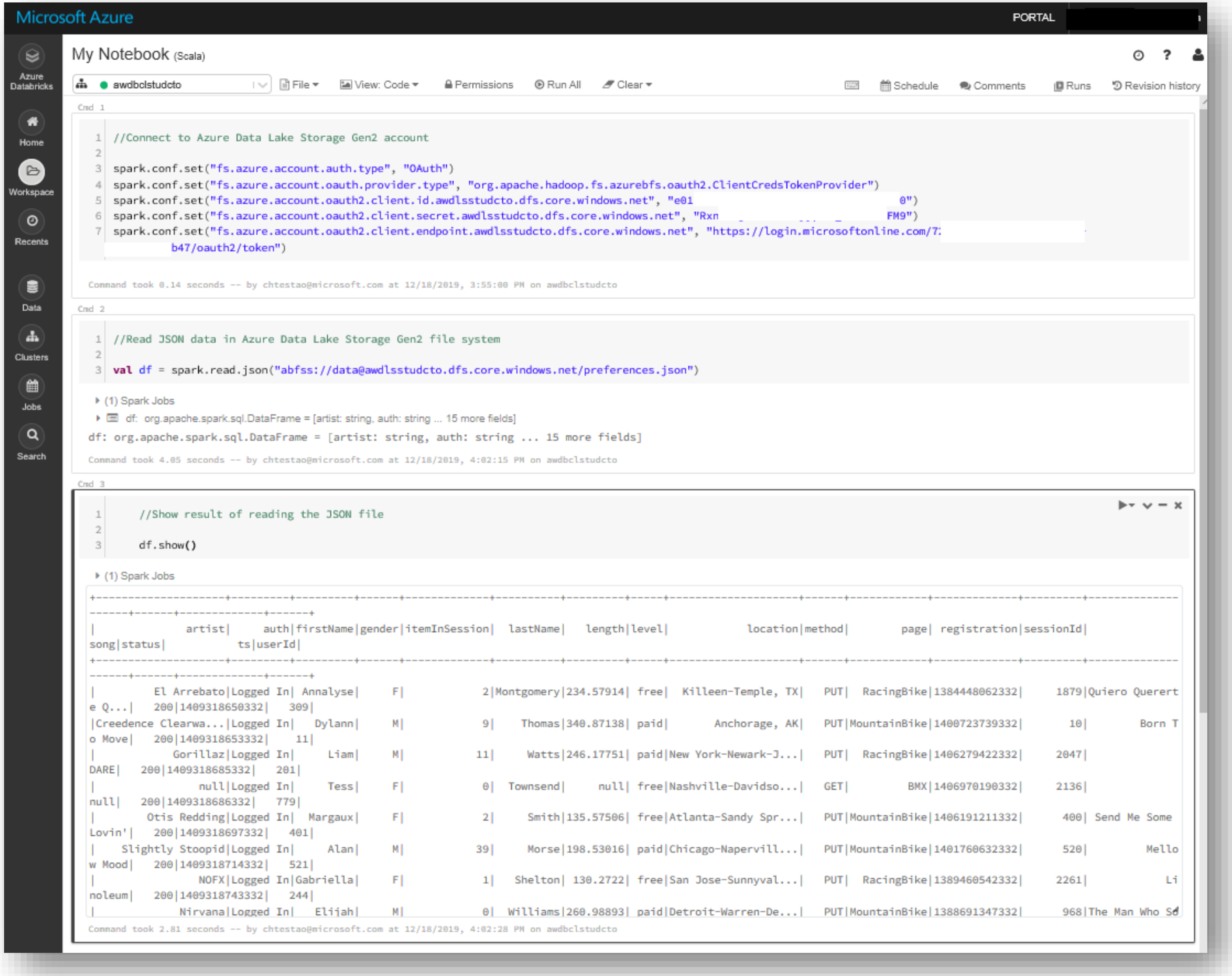
Select: DLAccess

DLAccess



```
//Connect to Azure Data Lake Storage Gen2 account

spark.conf.set("fs.azure.account.auth.type", "OAuth")
spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.awdlsstudcto.dfs.core.windows.net", "e01[redacted]210")
spark.conf.set("fs.azure.account.oauth2.client.secret.awdlsstudcto.dfs.core.windows.net", "Rx[redacted]M9")
spark.conf.set("fs.azure.account.oauth2.client.endpoint.awdlsstudcto.dfs.core.windows.net", "https://login.microsoftonline.com/72[redacted]
[redacted]db47/oauth2/token")
```

```
//Connect to Azure Data Lake Storage Gen2 account


spark.conf.set("fs.azure.account.auth.type", "OAuth")
spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.<storage-account-name>.dfs.core.windows.net", "<application-id>")
spark.conf.set("fs.azure.account.oauth2.client.secret.<storage-account-name>.dfs.core.windows.net", "<authentication-key>")
spark.conf.set("fs.azure.account.oauth2.client.endpoint.<storage-account-name>.dfs.core.windows.net",
"https://login.microsoftonline.com/<tenant-id>/oauth2/token")
```



Cmd 1

```
//Connect to Azure Data Lake Storage Gen2 account

spark.conf.set("fs.azure.account.auth.type", "OAuth")
spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.awdlsstudcto.dfs.core.windows.net", "e01                                0")
spark.conf.set("fs.azure.account.oauth2.client.secret.awdlsstudcto.dfs.core.windows.net", "Rxn                         FM9")
spark.conf.set("fs.azure.account.oauth2.client.endpoint.awdlsstudcto.dfs.core.windows.net", "https://login.microsoftonline.com/7:
          b47/oauth2/token")
```

Command took 0.14 seconds -- by chtestao@microsoft.com at 12/18/2019, 3:55:08 PM on awdbclstudcto

Cmd 2

```
//Read JSON data in Azure Data Lake Storage Gen2 file system

val df = spark.read.json("abfss://data@awdlsstudcto.dfs.core.windows.net/preferences.json")
```

```
▸ (1) Spark Jobs
▸ df: org.apache.spark.sql.DataFrame = [artist: string, auth: string ... 15 more fields]
df: org.apache.spark.sql.DataFrame = [artist: string, auth: string ... 15 more fields]
```

Command took 4.05 seconds -- by chtestao@microsoft.com at 12/18/2019, 4:02:15 PM on awdbclstudcto

Cmd 3

```
    //Show result of reading the JSON file

    df.show()
```

```
▸ (1) Spark Jobs
+--------------------+---------+---------+------+-------------+----------+---------+-----+--------------------+------+-------------+-------------+---------+------------
------+------+-------------+------+
|              artist|     auth|firstName|gender|itemInSession|  lastName|   length|level|            location|method|         page| registration|sessionId|
song|status|           ts|userId|
+--------------------+---------+---------+------+-------------+----------+---------+-----+--------------------+------+-------------+-------------+---------+------------
------+------+-------------+------+
|         El Arrebato|Logged In| Annalyse|     F|            2|Montgomery|234.57914| free|  Killeen-Temple, TX|   PUT|   RacingBike|1384448062332|     1879|Quiero Querert
e Q...|   200|1409318650332|   309|
|Creedence Clearwa...|Logged In|   Dylann|     M|            9|    Thomas|340.87138| paid|       Anchorage, AK|   PUT|MountainBike|1400723739332|       10|      Born T
o Move|   200|1409318653332|    11|
|             Gorillaz|Logged In|     Liam|     M|           11|     Watts|246.17751| paid|New York-Newark-J...|   PUT|   RacingBike|1406279422332|     2047|
DARE|   200|1409318685332|   201|
|                null|Logged In|     Tess|     F|            0|  Townsend|     null| free|Nashville-Davidso...|   GET|          BMX|1406970190332|     2136|
null|   200|1409318686332|   779|
|        Otis Redding|Logged In|  Margaux|     F|            2|     Smith|135.57506| free|Atlanta-Sandy Spr...|   PUT|MountainBike|1406191211332|      400| Send Me Some
Lovin'|   200|1409318697332|   401|
|     Slightly Stoopid|Logged In|     Alan|     M|           39|     Morse|198.53016| paid|Chicago-Napervill...|   PUT|MountainBike|1401760632332|      520|       Mello
w Mood|   200|1409318714332|   521|
|                 NOFX|Logged In|Gabriella|     F|            1|   Shelton| 130.2722| free|San Jose-Sunnyval...|   PUT|   RacingBike|1389460542332|     2261|          Li
noleum|   200|1409318743332|   244|
|              Nirvana|Logged In|   Elijah|     M|            0|  Williams|260.98893| paid|Detroit-Warren-De...|   PUT|MountainBike|1388691347332|      968|The Man Who Sö
```

Command took 2.81 seconds -- by chtestao@microsoft.com at 12/18/2019, 4:02:28 PM on awdbclstudcto

```
// Read JSON data in Azure Data Lake Storage Gen2
file system

val df = spark.read.json("abfss://<file-system-
name>@<storage-account-
name>.dfs.core.windows.net/preferences.json")

//Show result of reading the JSON file

df.show()
```

arif.works

Monitoring and Optimization

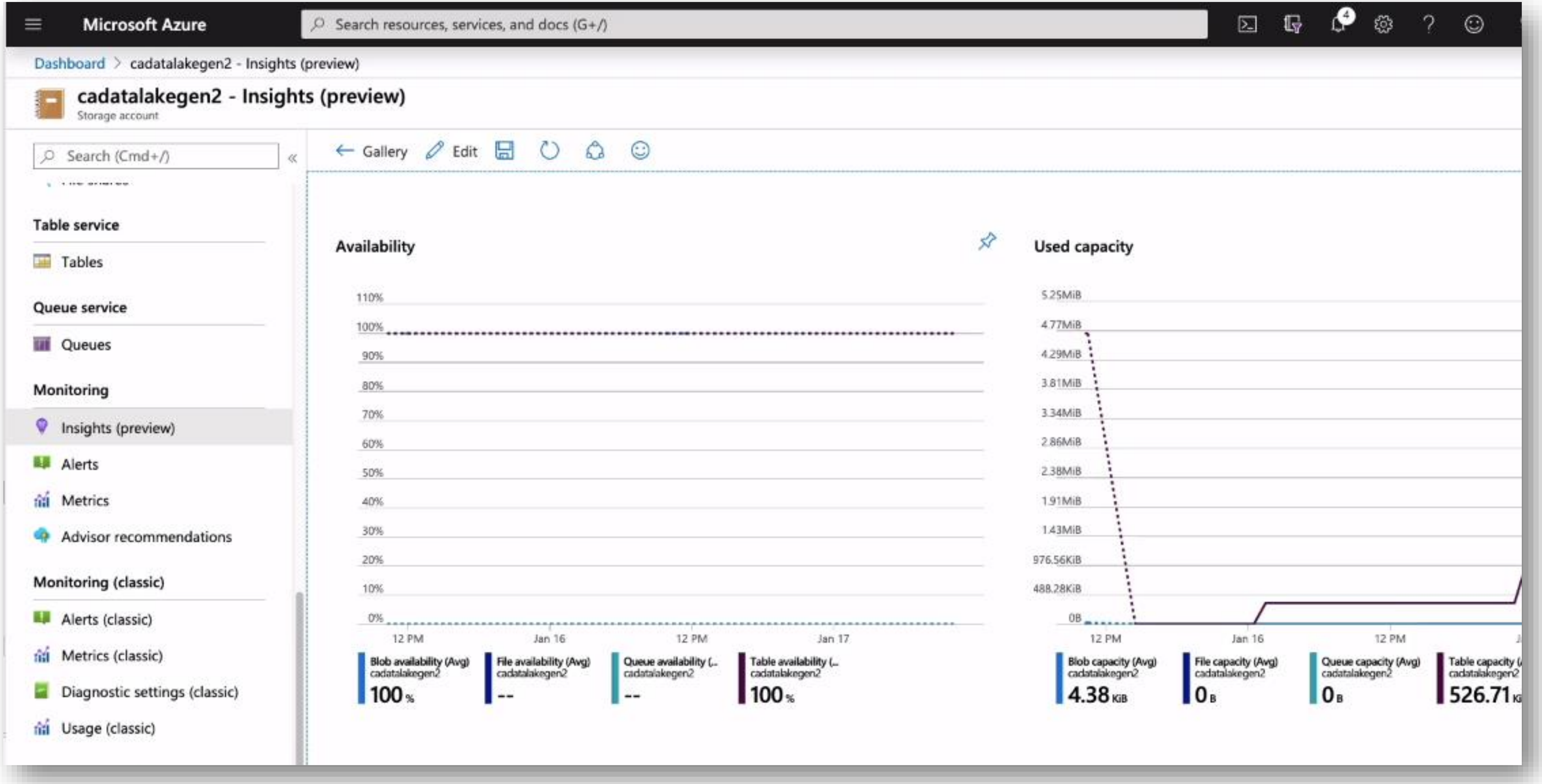
Monitoring and Optimization

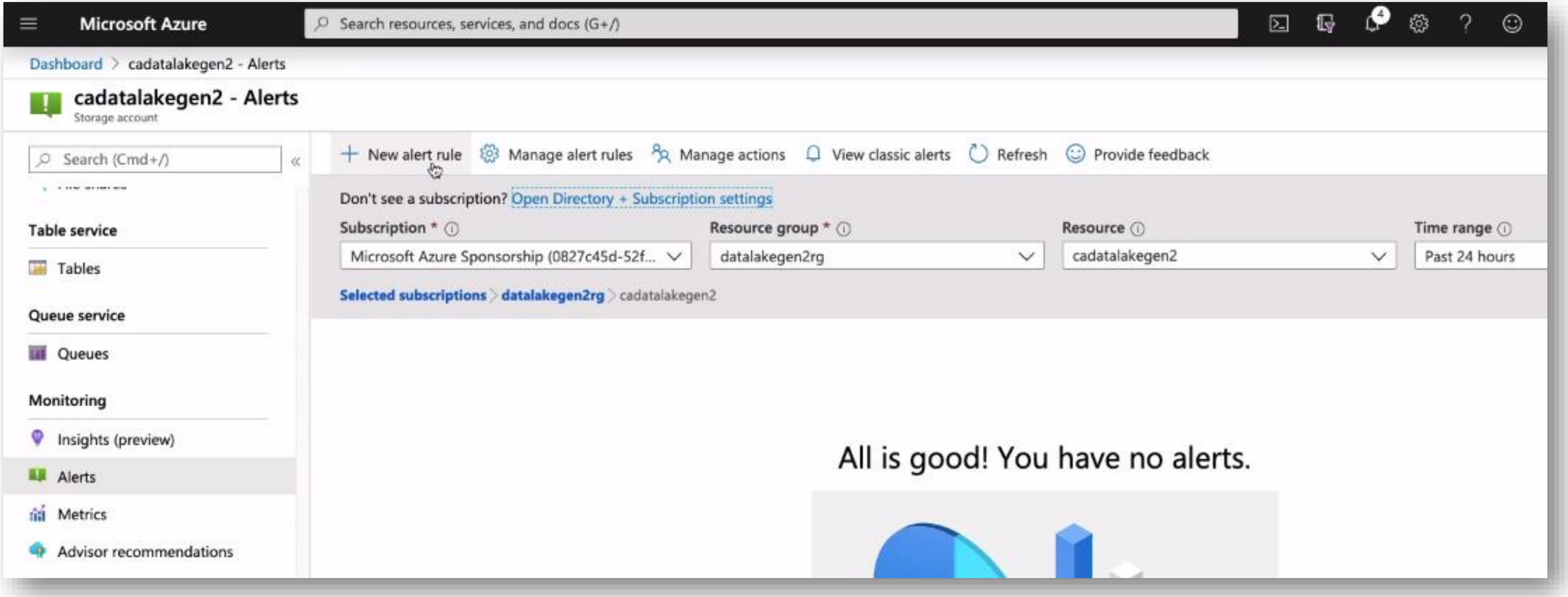
Monitoring

- **Azure Monitor** provides tools for keeping track of metrics and sending alerts

Optimization

- Use high-speed storage and networking for data ingestion


Microsoft Azure
Search resources, services, and docs (G+/)
Dashboard > cadatalakegen2 - Insights (preview)
cadatalakegen2 - Insights (preview)
Storage account
Search (Cmd+/)
Table service
Tables
Queue service
Queues
Monitoring
Insights (preview)
Alerts
Metrics
Advisor recommendations
Monitoring (classic)
Alerts (classic)
Metrics (classic)
Diagnostic settings (classic)
Usage (classic)
Gallery
Edit
Availability
110%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
12 PM
Jan 16
12 PM
Jan 17
Blob availability (Avg)
cadatalakegen2
100 %
File availability (Avg)
cadatalakegen2
--
Queue availability (...
cadatalakegen2
--
Table availability (...
cadatalakegen2
100 %
Used capacity
5.25MiB
4.77MiB
4.29MiB
3.81MiB
3.34MiB
2.86MiB
2.38MiB
1.91MiB
1.43MiB
976.56KiB
488.28KiB
0B
12 PM
Jan 16
12 PM
Blob capacity (Avg)
cadatalakegen2
4.38 KiB
File capacity (Avg)
cadatalakegen2
0 B
Queue capacity (Avg)
cadatalakegen2
0 B
Table capacity (
cadatalakegen2
526.71 Ki



Don't see a subscription? Open Directory + Subscription settings

Subscription * Microsoft Azure Sponsorship (0827c45d-52f...

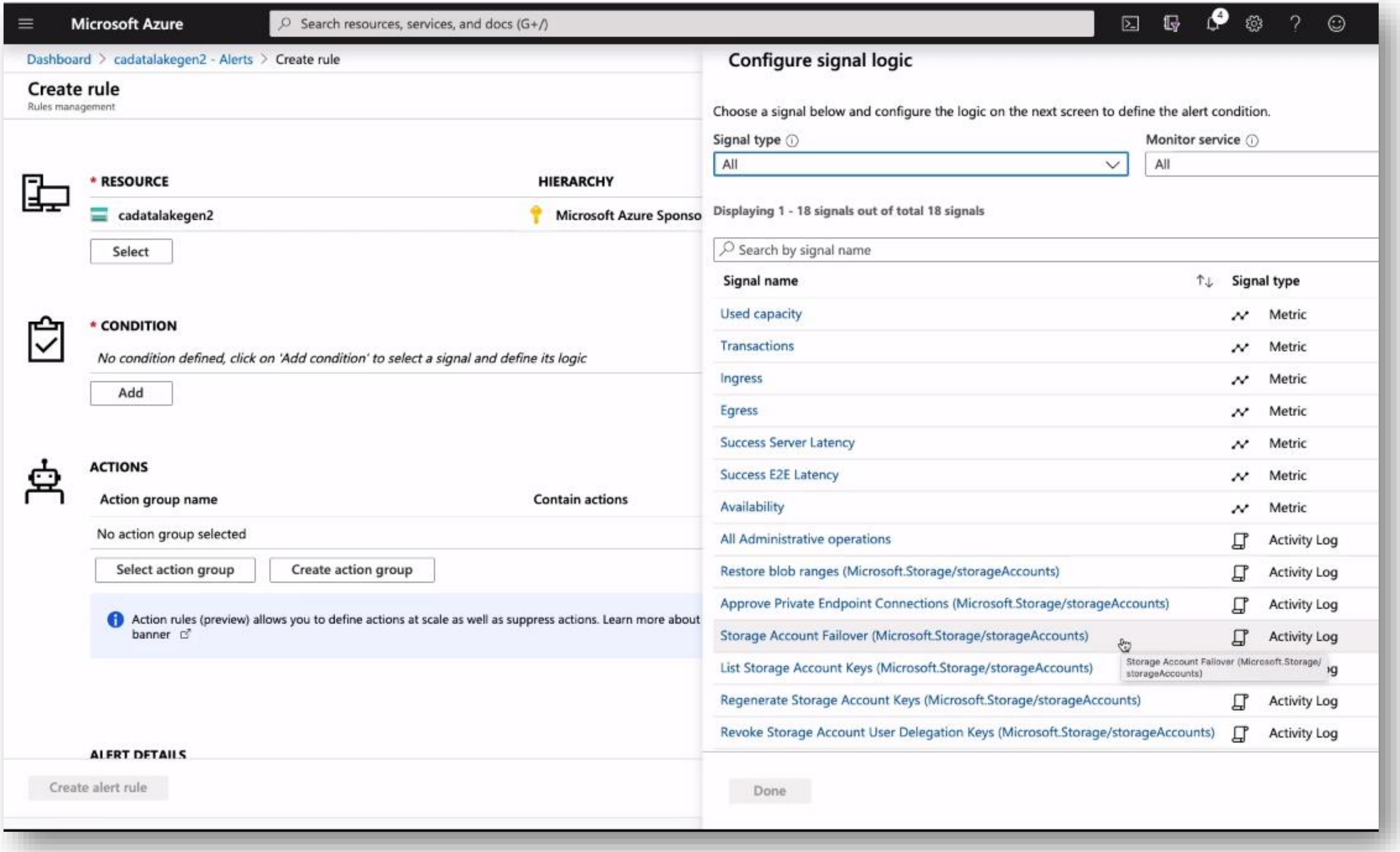
Resource group * datalakegen2rg

Resource cadatalakegen2

Time range Past 24 hours

Selected subscriptions > datalakegen2rg > cadatalakegen2

All is good! You have no alerts.



Dashboard > cadatalakegen2 - Alerts > Create rule

Create rule
Rules management

*** RESOURCE** | **HIERARCHY**

cadatalakegen2 | Microsoft Azure Sponso

*** CONDITION**

No condition defined, click on 'Add condition' to select a signal and define its logic

ACTIONS

Action group name	Contain actions
No action group selected	

Action rules (preview) allows you to define actions at scale as well as suppress actions. Learn more about banner

ALERT DETAILS

Configure signal logic

Choose a signal below and configure the logic on the next screen to define the alert condition.

Signal type: All

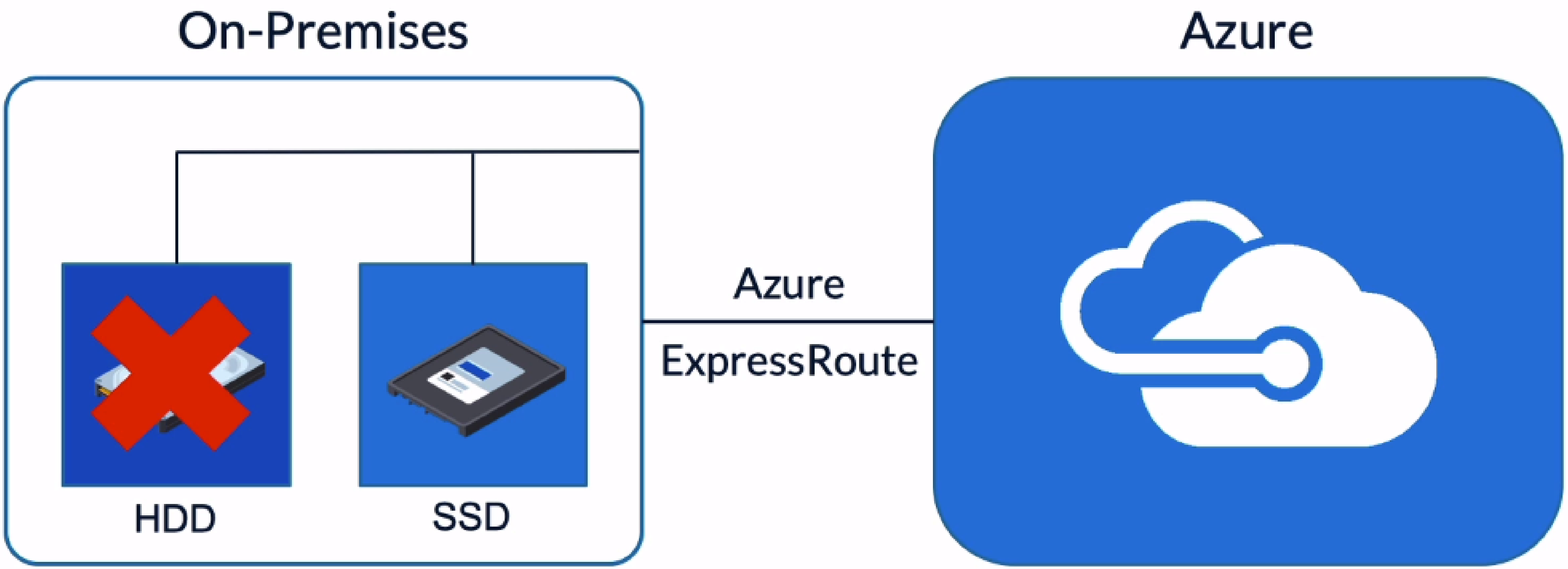
Monitor service: All

Displaying 1 - 18 signals out of total 18 signals

Signal name	Signal type
Used capacity	Metric
Transactions	Metric
Ingress	Metric
Egress	Metric
Success Server Latency	Metric
Success E2E Latency	Metric
Availability	Metric
All Administrative operations	Activity Log
Restore blob ranges (Microsoft.Storage/storageAccounts)	Activity Log
Approve Private Endpoint Connections (Microsoft.Storage/storageAccounts)	Activity Log
Storage Account Failover (Microsoft.Storage/storageAccounts)	Activity Log
List Storage Account Keys (Microsoft.Storage/storageAccounts)	ig
Regenerate Storage Account Keys (Microsoft.Storage/storageAccounts)	Activity Log
Revoke Storage Account User Delegation Keys (Microsoft.Storage/storageAccounts)	Activity Log

Storage Account Failover (Microsoft.Storage/storageAccounts)
Done

Data Ingestion



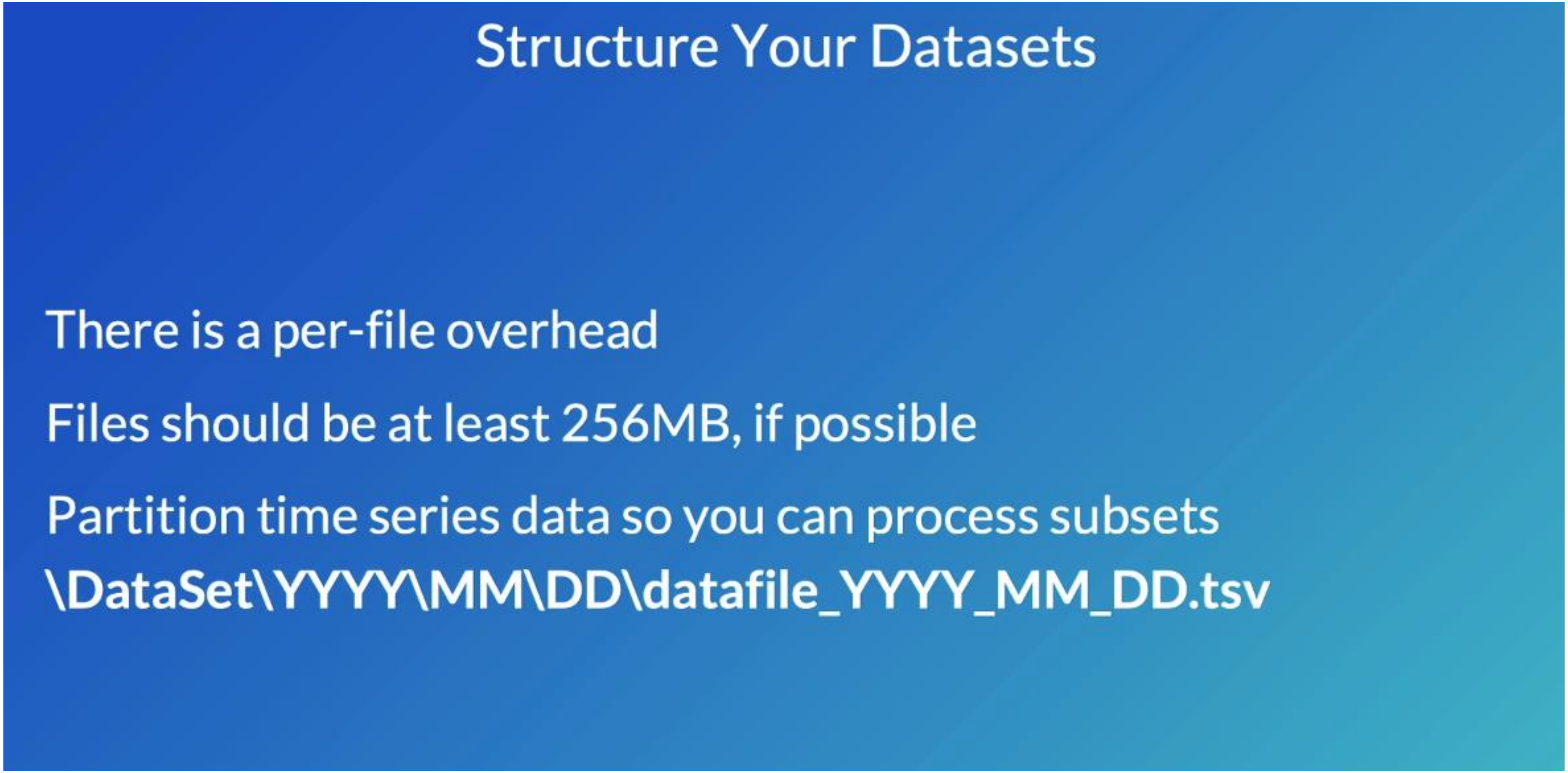
Configure data ingestion tools for maximum parallelization

When you are transferring large amounts of data from your local infrastructure to the Azure cloud, there are three potential bottlenecks.

First is the speed of your storage. Ideally, your local data should be on SSDs rather than spinning disks, and on storage arrays rather than individual disks.

Second, you should have a high-speed internal network. In particular, the network interface cards on your local machines should be as fast as possible.

Third, the network connection between your local infrastructure and the Azure cloud should be fast. If it’ s a major bottleneck, then consider using a dedicated link with Azure ExpressRoute.



Structure Your Datasets

There is a per-file overhead

Files should be at least 256MB, if possible

Partition time series data so you can process subsets

\DataSet\YYYY\MM\DD\datafile_YYYY_MM_DD.tsv

Cosmos DB

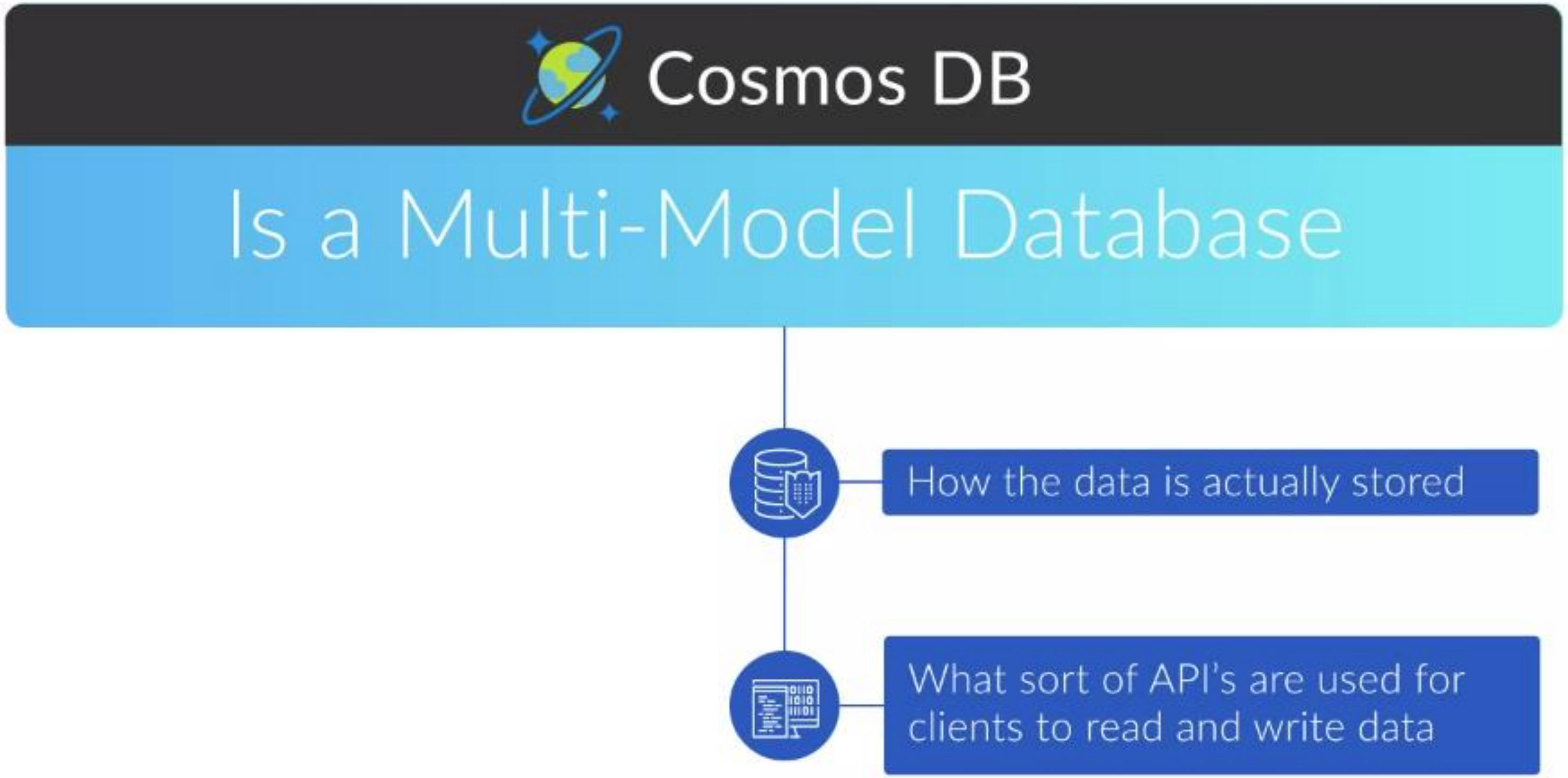
Introduction to Cosmos DB

Cosmos DB

- An extension of Microsoft's DocumentDB
- Availability is one of the chief guarantees
  - Achieving reliable database performance across multiple geographic regions is very difficult

Cosmos DB is a globally distributed multi-model database

Introduction to Cosmos DB



Cosmos DB

Is a Multi-Model Database

How the data is actually stored

What sort of API's are used for clients to read and write data

Introduction to Cosmos DB

Cosmos DB

- It's a database as a service.
- Sign up for access through Azure and behind the scenes Microsoft handles the scaling.
- Microsoft Azure offers a **99.999% uptime** SLA to ease concerns about entrusting your data to them.

Keep in mind the relative lack of control you have if you rely on Cosmos DB. It isn't like a typical database system where you can SSH to the server and tune things the way you want.

Cosmos DB

Cosmos DB is a globally distributed multi-model database

That’s it in a nutshell

Cosmos DB is a powerful database as a service system designed to support a variety of data models and work across multiple geographic regions.

Cosmos DB Features and Capabilities

We’re going to focus on the **six most important and compelling capabilities**.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Global Distribution of Data

READ ONLY
WRITE ONLY
READ/WRITE

It was designed from the ground up to support access patterns from all over the planet. With over 50 geographic locations for its data centers, Cosmos DB users can ensure minimal latency for their users.

New locations are regularly added each year.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Global Distribution of Data

READ ONLY
WRITE ONLY
READ/WRITE
FAILOVER PRIORITY
LOW
HIGH

It was designed from the ground up to support access patterns from all over the planet. With over 50 geographic locations for its data centers, Cosmos DB users can ensure minimal latency for their users.

New locations are regularly added each year.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Serverless Architecture

Cosmos DB is an example of database as a service.

You do not set up database servers. You get an endpoint for your app to utilize.

Costs are determined using **Request Units**.

The larger your data, the more frequent your queries, the more indexing you do, the more consistency you demand, the larger the number of request units you will need.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Serverless Architecture

This system greatly simplifies your data layer.

You don't need to think about memory, CPU, hardware provisioning, OS optimization, updates, patches, SSL certs, etc. All of the operational overhead of managing a database is gone. The time saved can translate into more than enough cost-savings to offset the cost of the needed request units.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Serverless Architecture

Cosmos DB's serverless architecture also ensures strong SLA's. You get a guarantee of **99.999%** uptime. You also get first order integration with other Azure services.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Multi-Model Support

Table Key-Value API

Cosmos DB offers **API's** for **Cassandra, Gremlin, MongoDB, SQL**, and **Table key-value API.** This means that with a single Cosmos DB account, you can run multiple database engines.

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Throughput + Consistency Guarantees

Cosmos DB is very strong on partition tolerance and availability.
Cosmos DB features five different consistency settings:

- **Strong**
- **Bounded Staleness**
- **Session**
- **Consistent Prefix**
- **Eventual**

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Throughput + Consistency Guarantees

Cosmos DB is very strong on partition tolerance and availability. Cosmos DB features five different consistency settings:

Strong
Ensures that no reads are processed until writes are completed durably by a quorum of replicas

Bounded Staleness
You get a configurable level of consistency. Reads will lag behind writes by either an adjustable time interval or a number of item revisions.

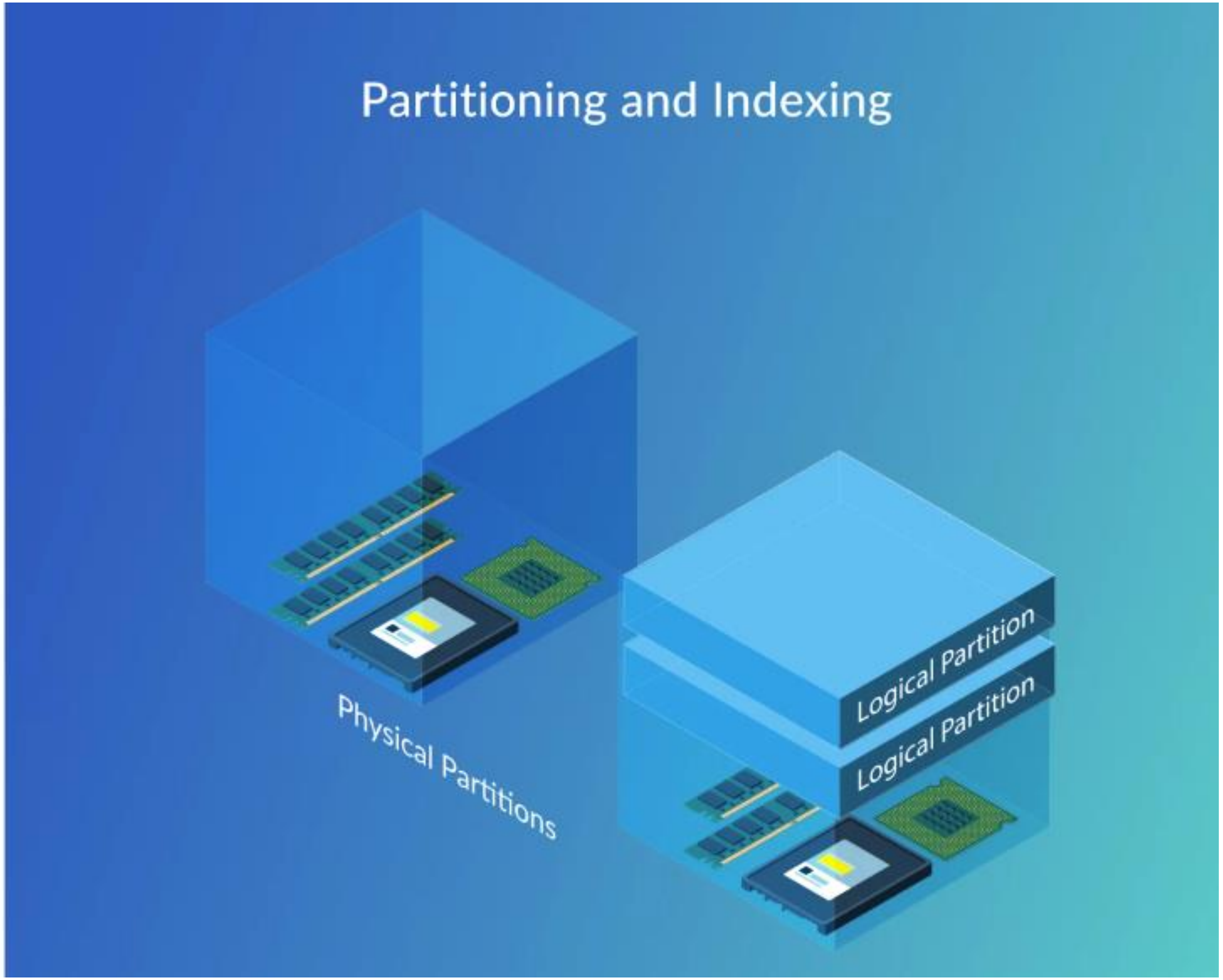
Session
Gives you a "read your own writes" guarantee suitable for scenarios where you need guarantees at the level of individual clients.

Consistent Prefix
Guarantees that your data will eventually converge to the most recently written. You get a guarantee that data will never be out of order.

Eventual
Guarantees that your data will eventually converge to the most recently written.

The 6 Features Are:

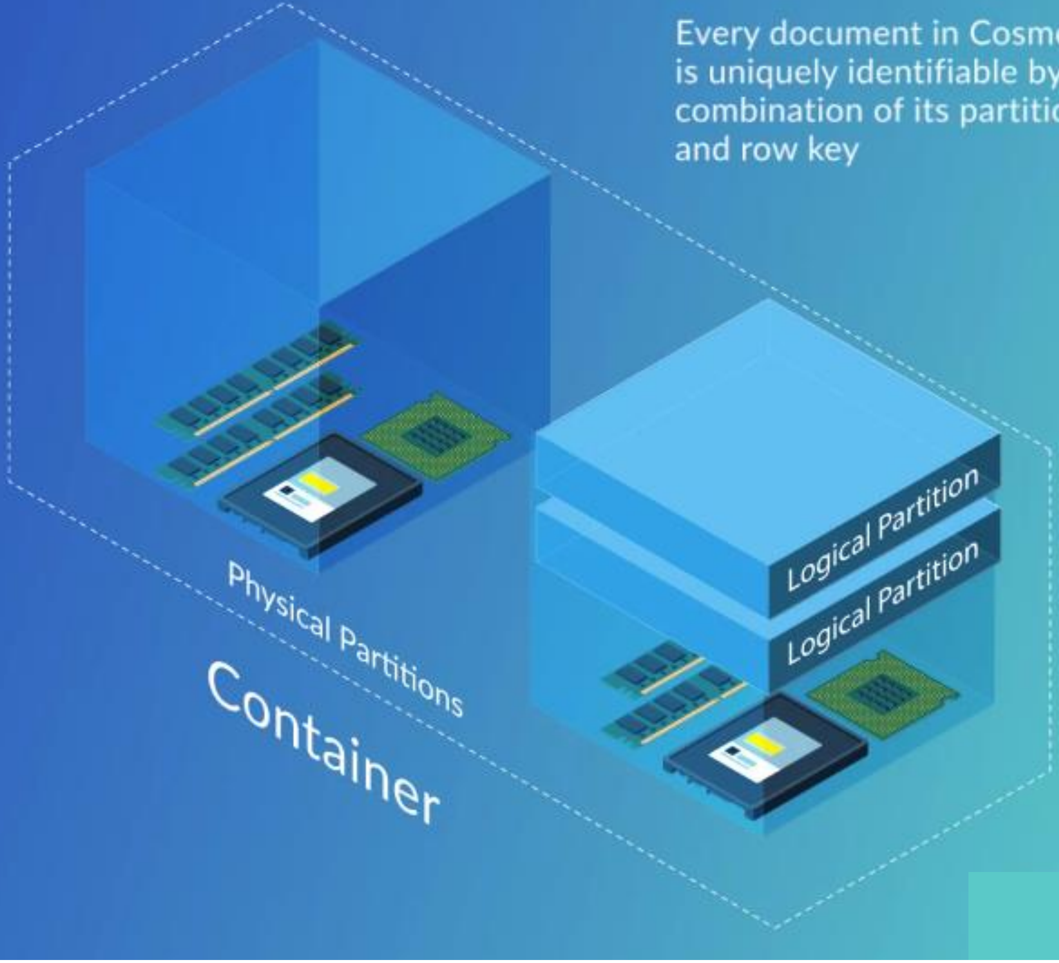
- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security



The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Partitioning and Indexing



Every document in Cosmos DB is uniquely identifiable by the combination of its partition key and row key

The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Partitioning and Indexing

When designing collections of data, two critical things to think about are partition key and indexing.

- Cosmos DB automatically indexes all of your data
- It is possible to create custom indexing policies that let you tune tradeoffs between query throughput and consistency
- The main thing you want is a column with high cardinality and a large variety of values to help distribute your workload evenly

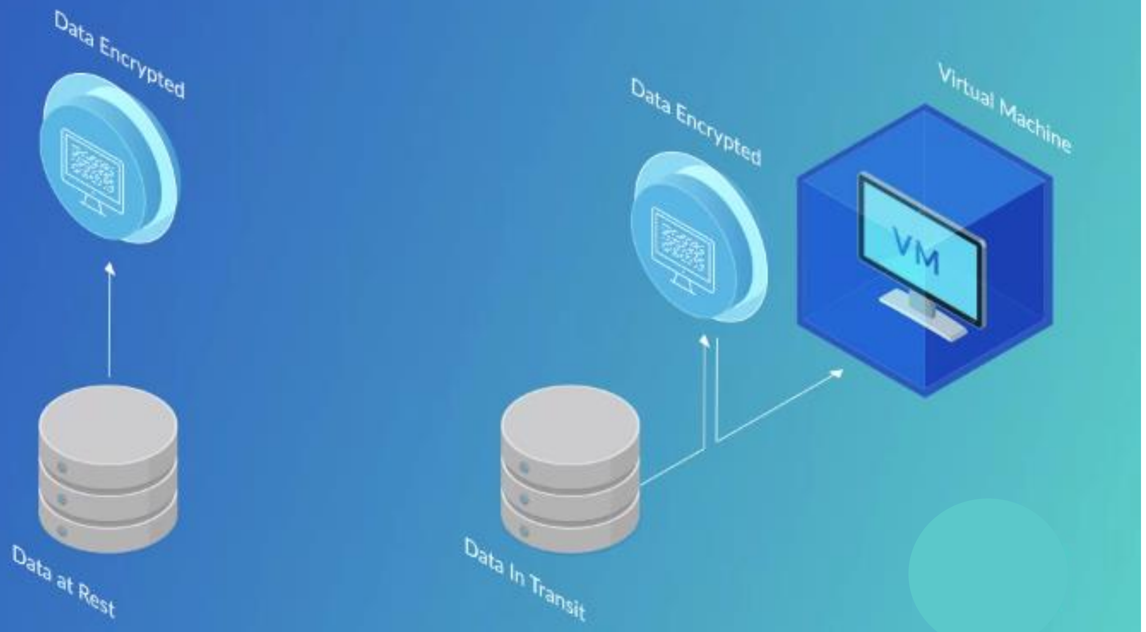
The 6 Features Are:

- Global Distribution of Data
- Serverless Architecture
- Multi-Model Support
- Throughput + Consistency Guarantees
- Partitioning and Indexing
- Security

Security

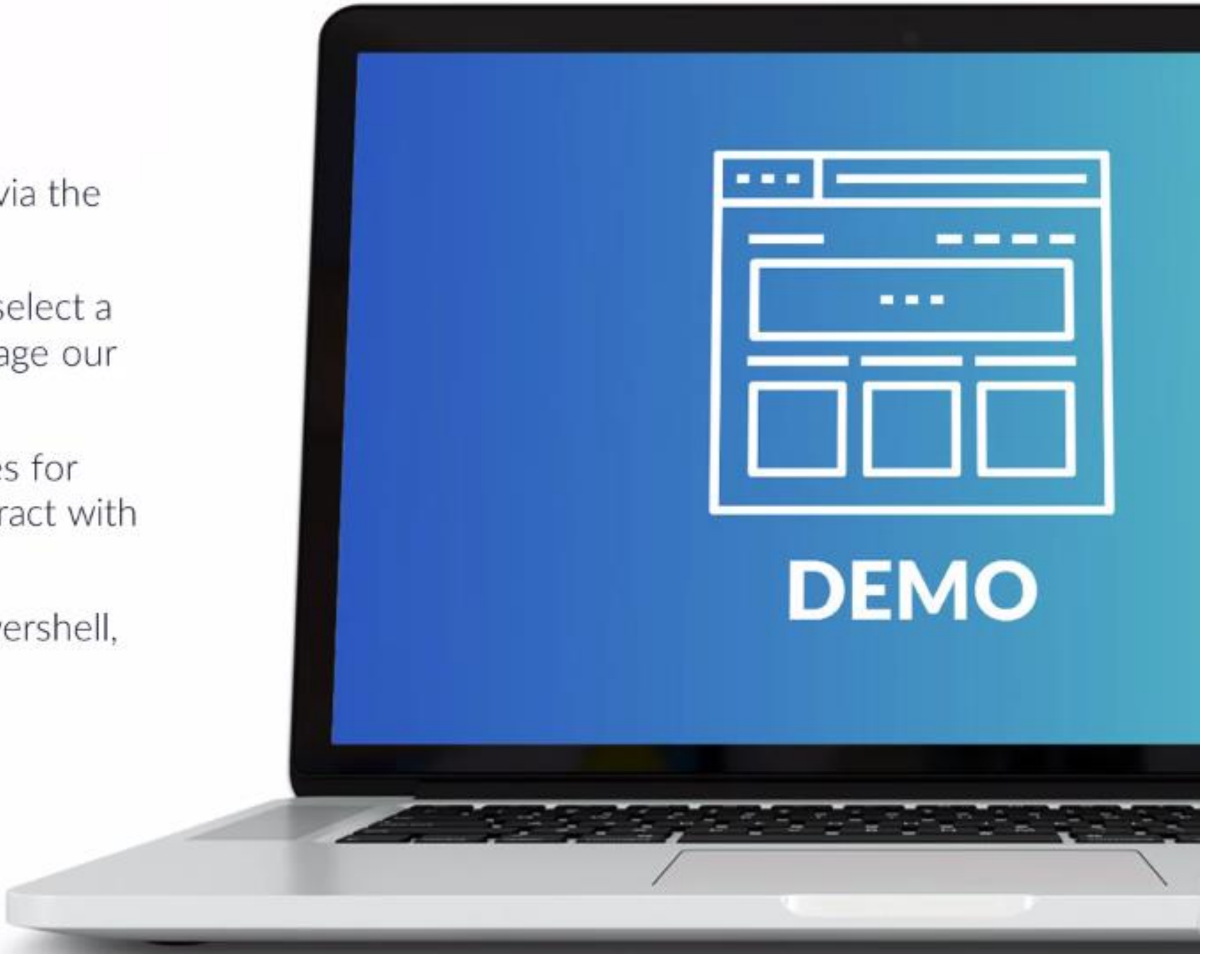
Cosmos DB has many of the same security considerations as any other provider. You need to control:

- Who has access to your Azure account
- Who has credentials to use the API
- Ensure that sensitive data is properly isolated



Introduction to Using Cosmos DB

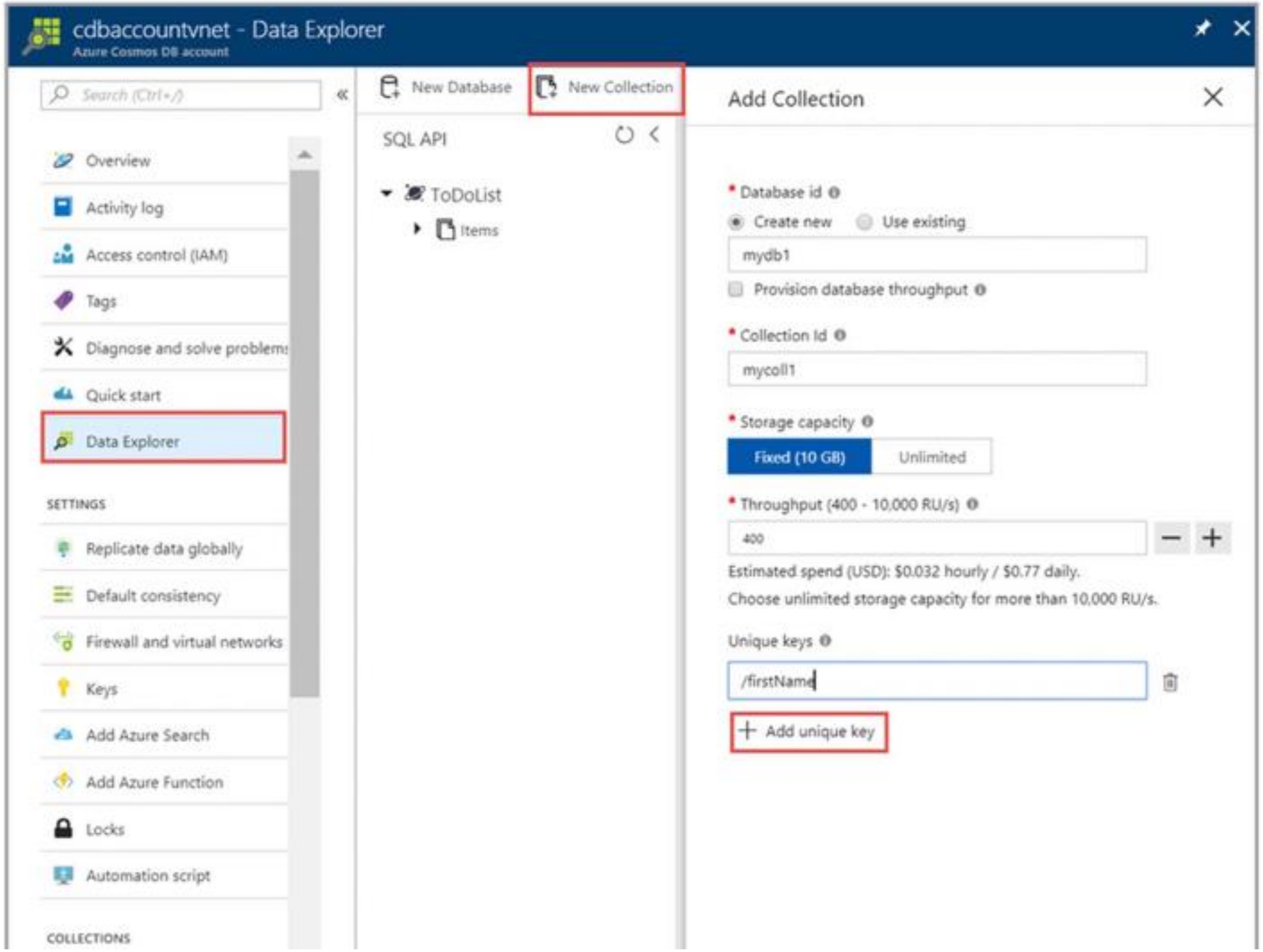
- We will start with just the initial setup via the web console.
- We will create a Cosmos DB account, select a data model API, and show how to manage our data via the browser.
- Next we will explain how to use libraries for languages like Python and .NET to interact with our Cosmos DB storage.
- We will also cover using CLI tools, Powershell, and REST API for Cosmos DB.



Configure unique keys by using Azure Portal

In the sections above you'll find code samples that will show how you can define unique key constraints when a collection is created using the SQL API or MongoDB API. But it's also possible to define unique keys when you create a collection via the web UI in the Azure portal.

- Navigate to the **Data Explorer** in your Cosmos DB account
- Click **New Collection**
- In the section Unique keys,** you can add the desired unique key constraints by clicking **Add unique key**



* Database id

● Create new ○ Use existing

mydb1

□ Provision database throughput

* Collection Id

mycoll1

* Storage capacity

[✓ Fixed (10 GB)] Unlimited

* Throughput (400 - 10,000 RU/s)

400

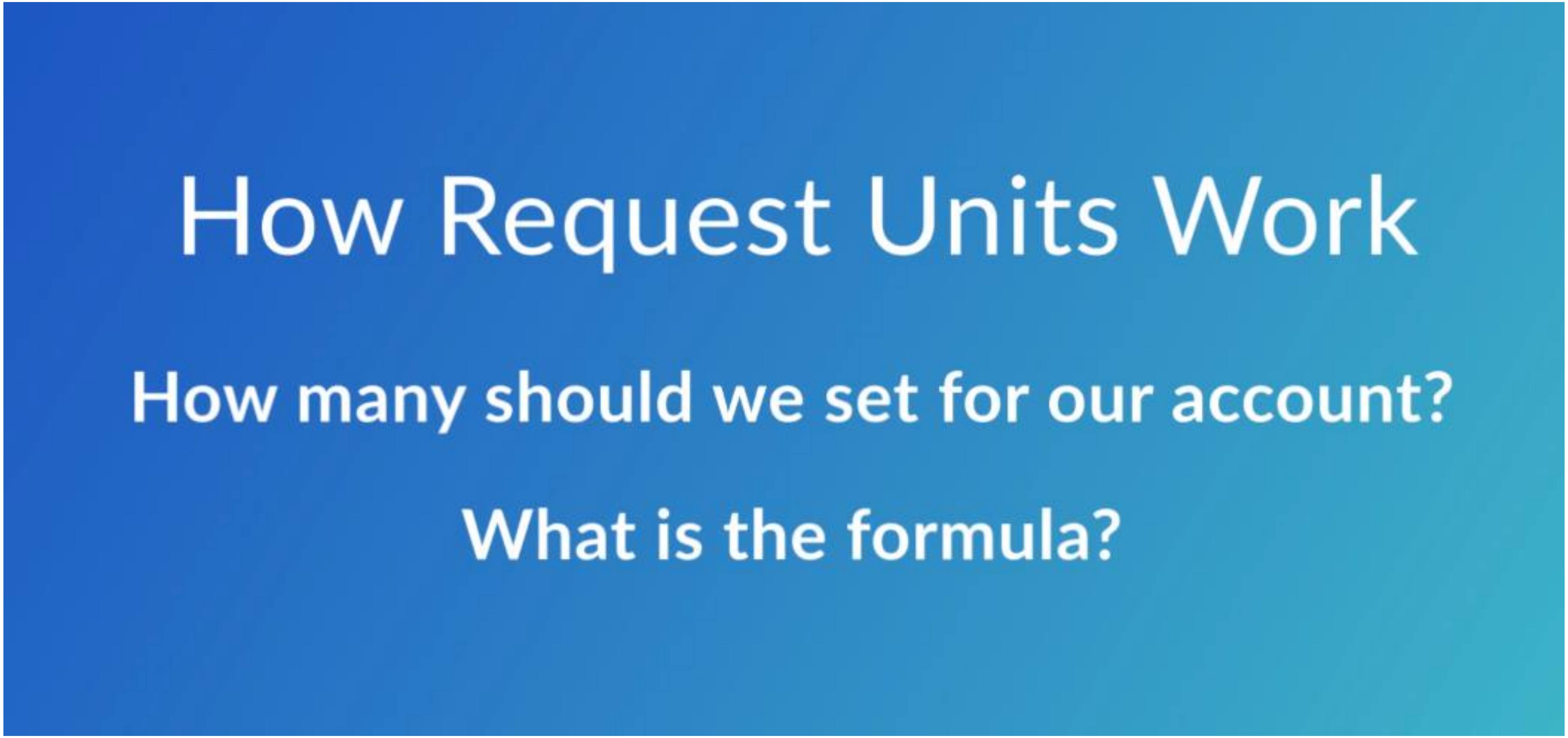
Estimated spend (USD): $0.032 hourly / $0.77 daily.

Choose unlimited storage capacity for more than 10,000 RU/s.

Unique keys

/firstName

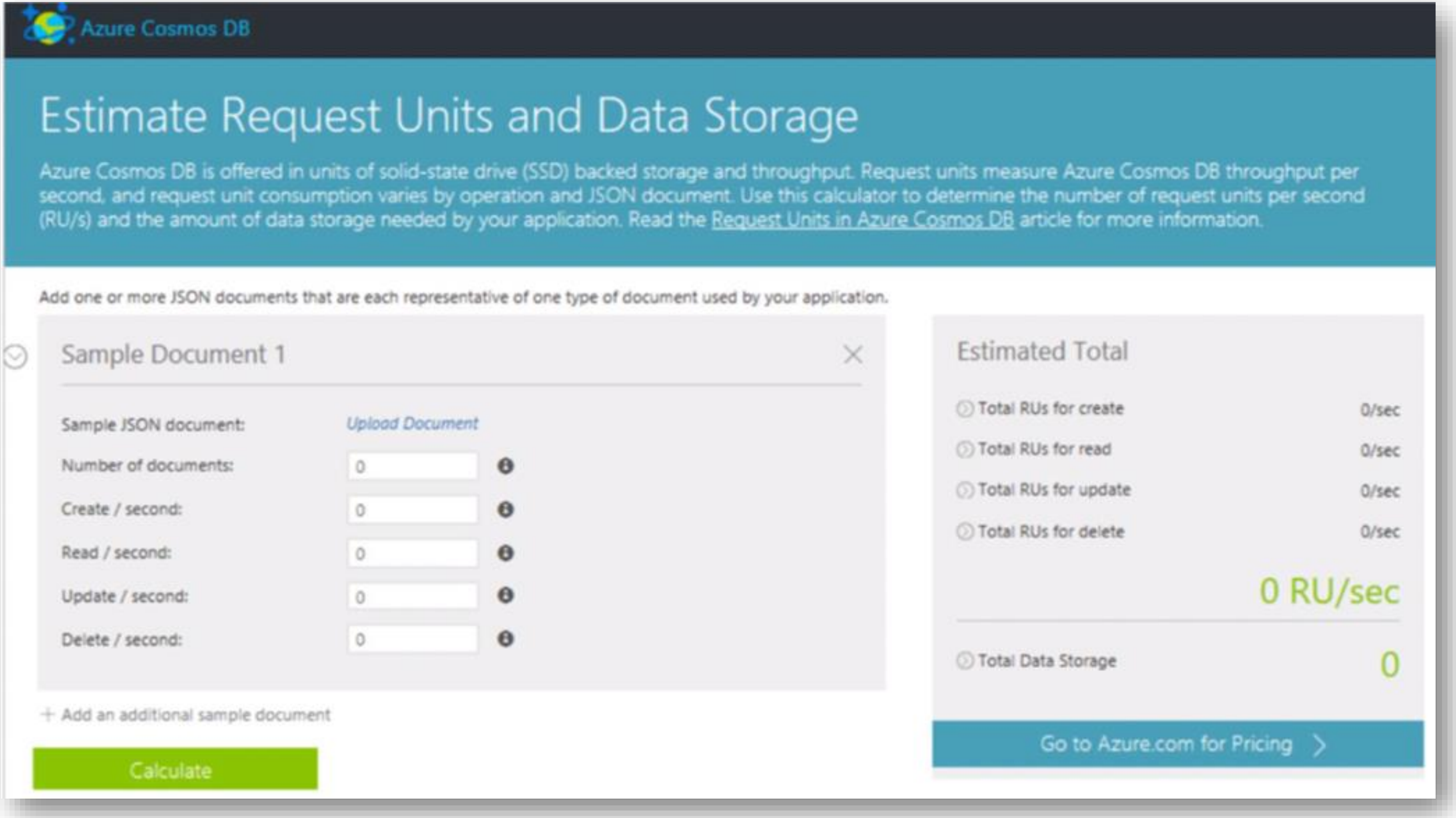
+ Add unique key



How Request Units Work

How many should we set for our account?

What is the formula?



Azure Cosmos DB

Estimate Request Units and Data Storage

Azure Cosmos DB is offered in units of solid-state drive (SSD) backed storage and throughput. Request units measure Azure Cosmos DB throughput per second, and request unit consumption varies by operation and JSON document. Use this calculator to determine the number of request units per second (RU/s) and the amount of data storage needed by your application. Read the Request Units in Azure Cosmos DB article for more information.

Add one or more JSON documents that are each representative of one type of document used by your application.

Sample Document 1

Sample JSON document:	*Upload Document*
Number of documents:	0
Create / second:	0
Read / second:	0
Update / second:	0
Delete / second:	0

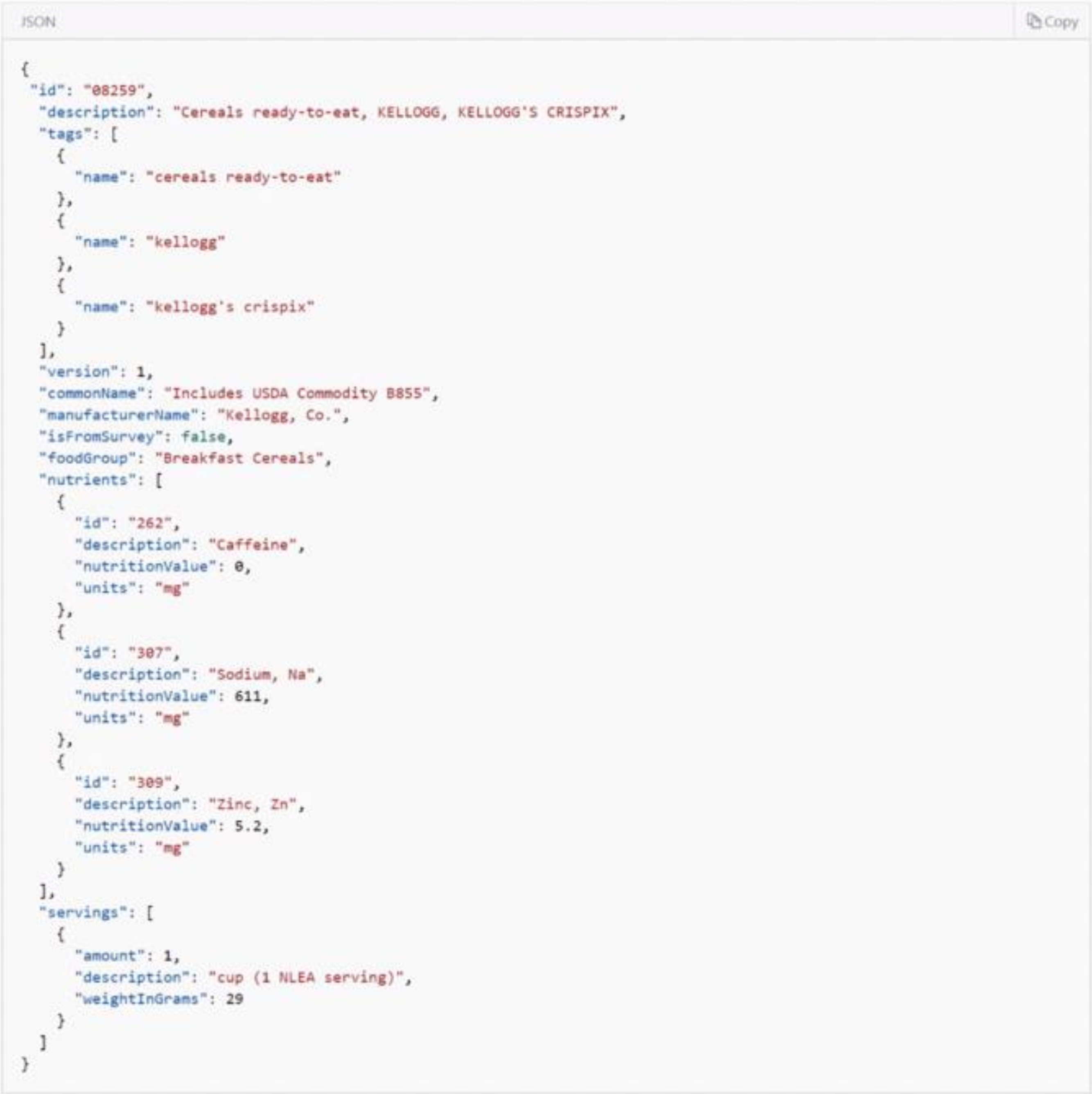
+ Add an additional sample document

Calculate

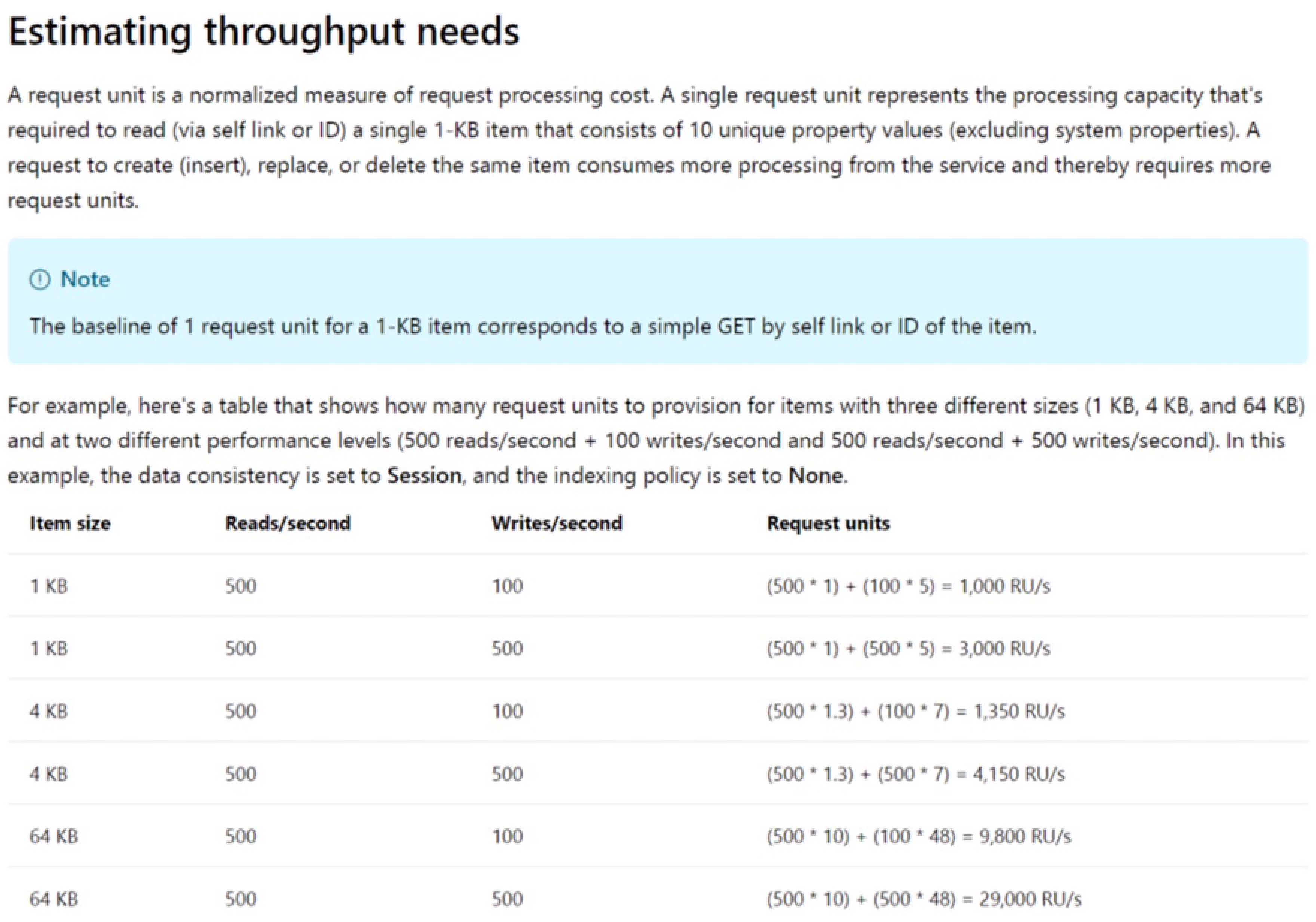
Estimated Total

Total RUs for create	0/sec
Total RUs for read	0/sec
Total RUs for update	0/sec
Total RUs for delete	0/sec
	0 RU/sec
Total Data Storage	0

Go to Azure.com for Pricing >



```
{
 "id": "08259",
  "description": "Cereals ready-to-eat, KELLOGG, KELLOGG'S CRISPIX",
  "tags": [
    {
      "name": "cereals ready-to-eat"
    },
    {
      "name": "kellogg"
    },
    {
      "name": "kellogg's crispix"
    }
  ],
  "version": 1,
  "commonName": "Includes USDA Commodity B855",
  "manufacturerName": "Kellogg, Co.",
  "isFromSurvey": false,
  "foodGroup": "Breakfast Cereals",
  "nutrients": [
    {
      "id": "262",
      "description": "Caffeine",
      "nutritionValue": 0,
      "units": "mg"
    },
    {
      "id": "307",
      "description": "Sodium, Na",
      "nutritionValue": 611,
      "units": "mg"
    },
    {
      "id": "309",
      "description": "Zinc, Zn",
      "nutritionValue": 5.2,
      "units": "mg"
    }
  ],
  "servings": [
    {
      "amount": 1,
      "description": "cup (1 NLEA serving)",
      "weightInGrams": 29
    }
  ]
}
```



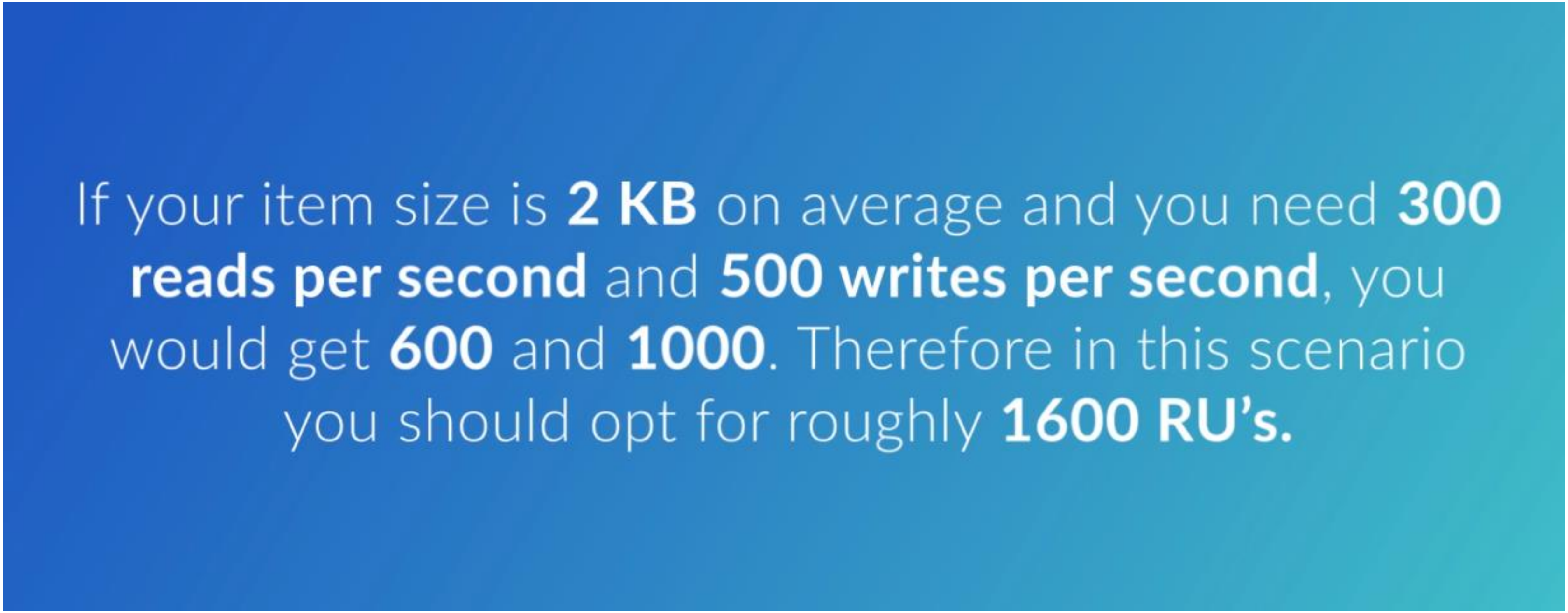
Estimating throughput needs

A request unit is a normalized measure of request processing cost. A single request unit represents the processing capacity that's required to read (via self link or ID) a single 1-KB item that consists of 10 unique property values (excluding system properties). A request to create (insert), replace, or delete the same item consumes more processing from the service and thereby requires more request units.

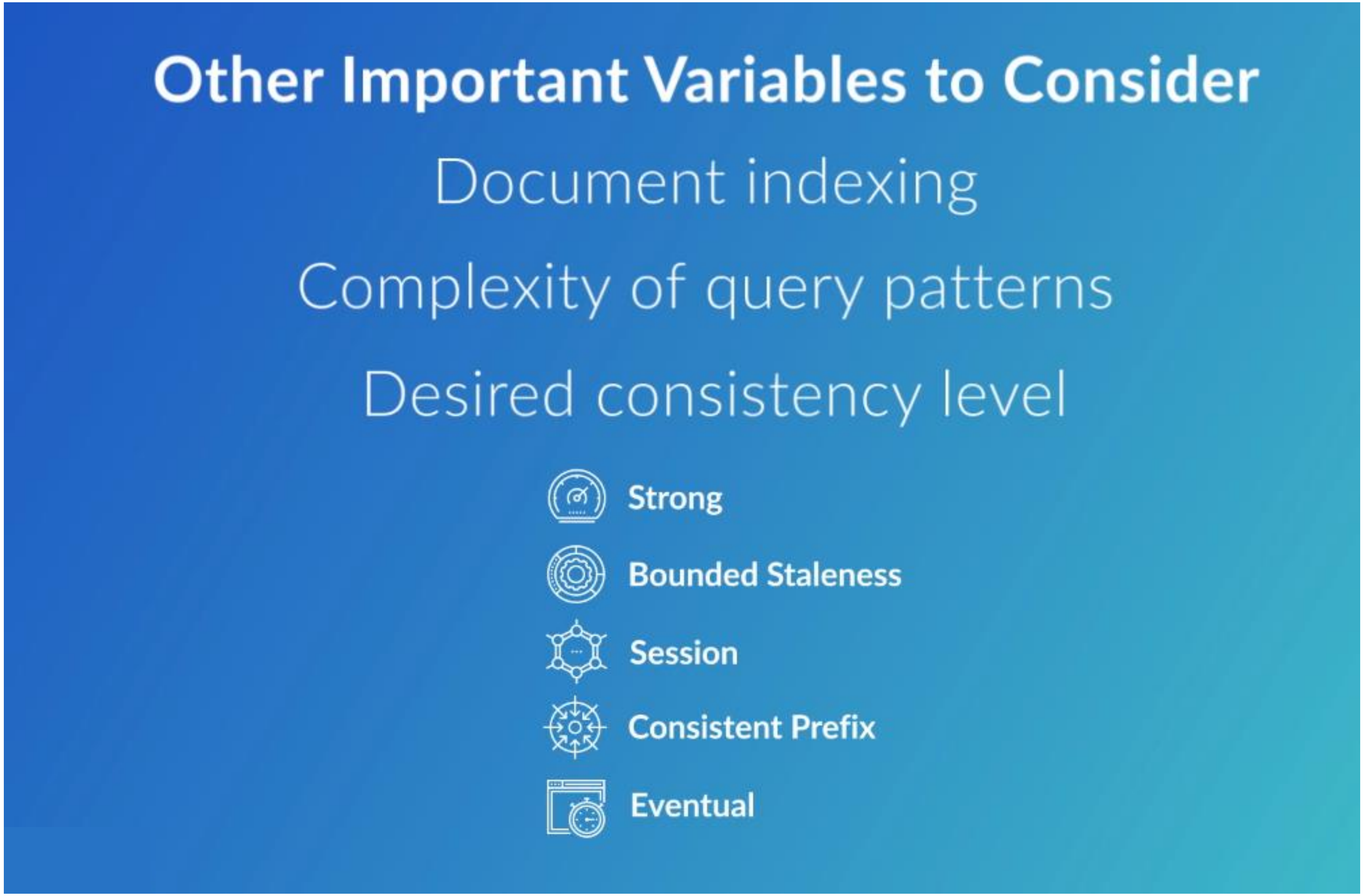
> ⓘ **Note**
>
> The baseline of 1 request unit for a 1-KB item corresponds to a simple GET by self link or ID of the item.

For example, here's a table that shows how many request units to provision for items with three different sizes (1 KB, 4 KB, and 64 KB) and at two different performance levels (500 reads/second + 100 writes/second and 500 reads/second + 500 writes/second). In this example, the data consistency is set to **Session**, and the indexing policy is set to **None**.

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



If your item size is **2 KB** on average and you need **300 reads per second** and **500 writes per second**, you would get **600** and **1000**. Therefore in this scenario you should opt for roughly **1600 RU's.**



Other Important Variables to Consider

Document indexing

Complexity of query patterns

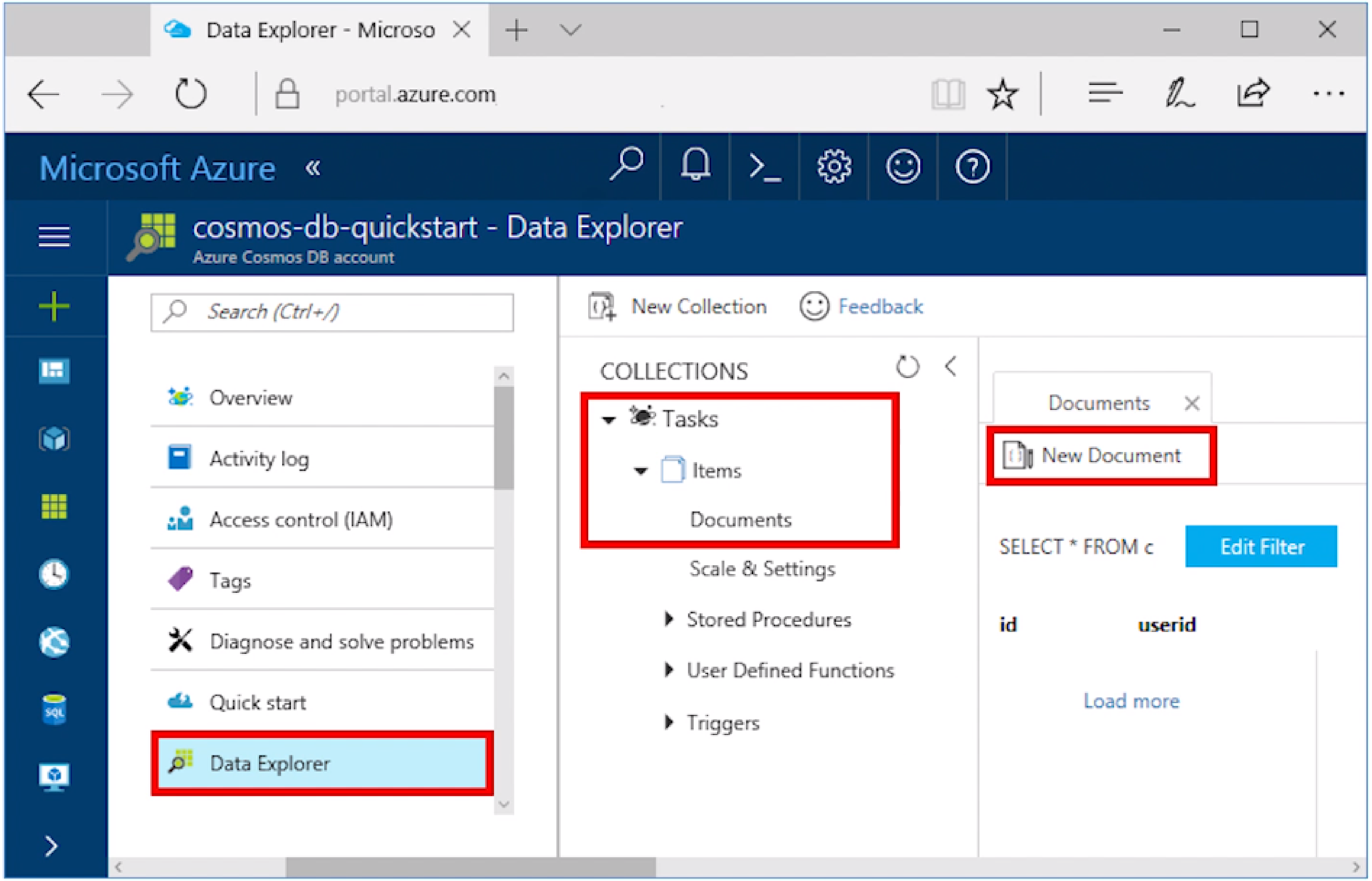
Desired consistency level

- Strong
- Bounded Staleness
- Session
- Consistent Prefix
- Eventual

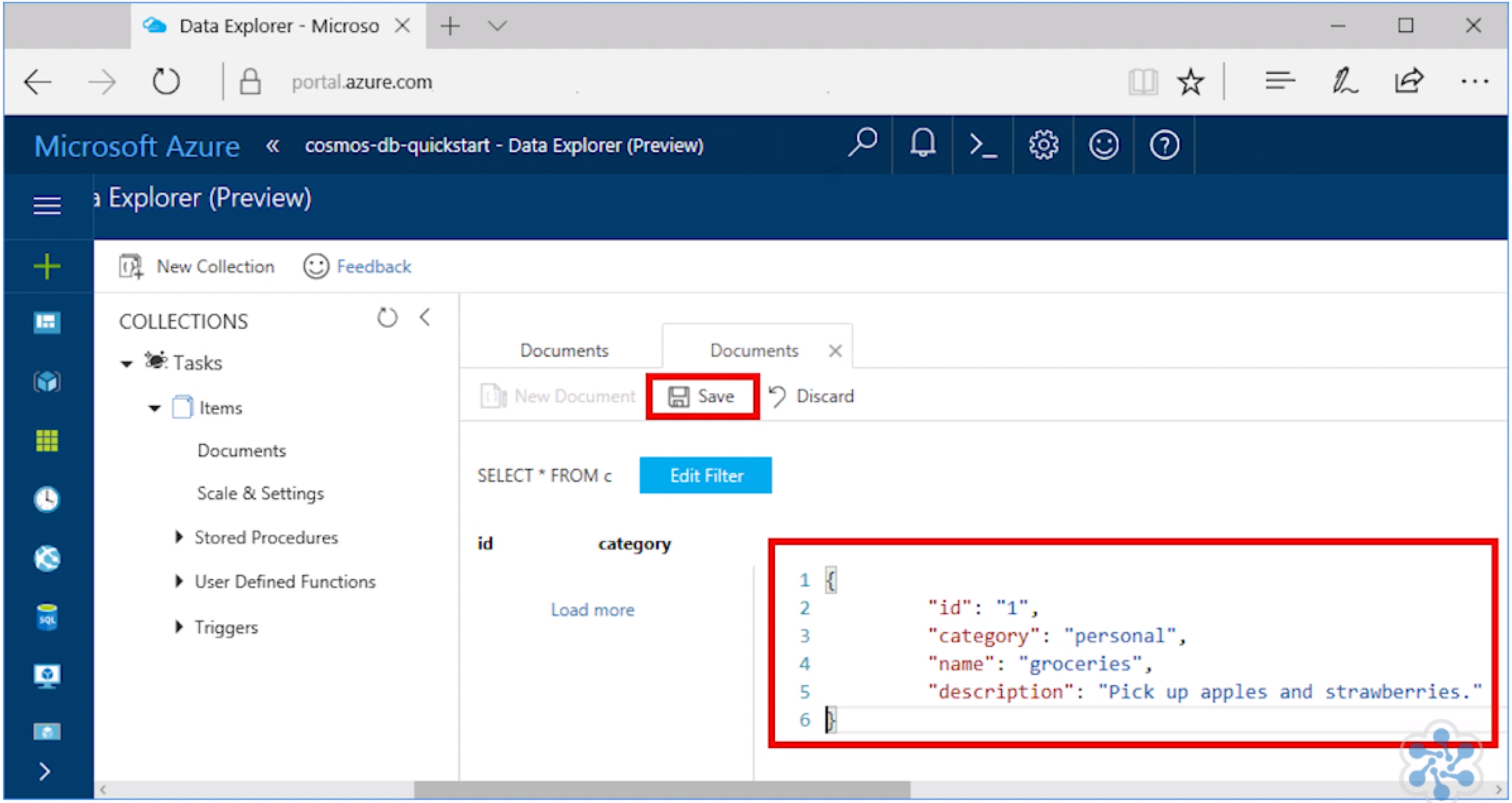
Backups

- Cosmos DB takes snapshots every **four hours** and stores them in geo-redundant Blob storage
- Only the last two snapshots are retained
- Snapshot of deleted databases or containers will be retained for **30 days**
- You can schedule additional backups using the Cosmos DB Data Migration tool
- Submit a ticket with Azure Support to restore accidentally deleted databases and containers
- If you have a data corruption issue, delete the corrupted container as soon as possible

Backup



```
SELECT * FROM c
```



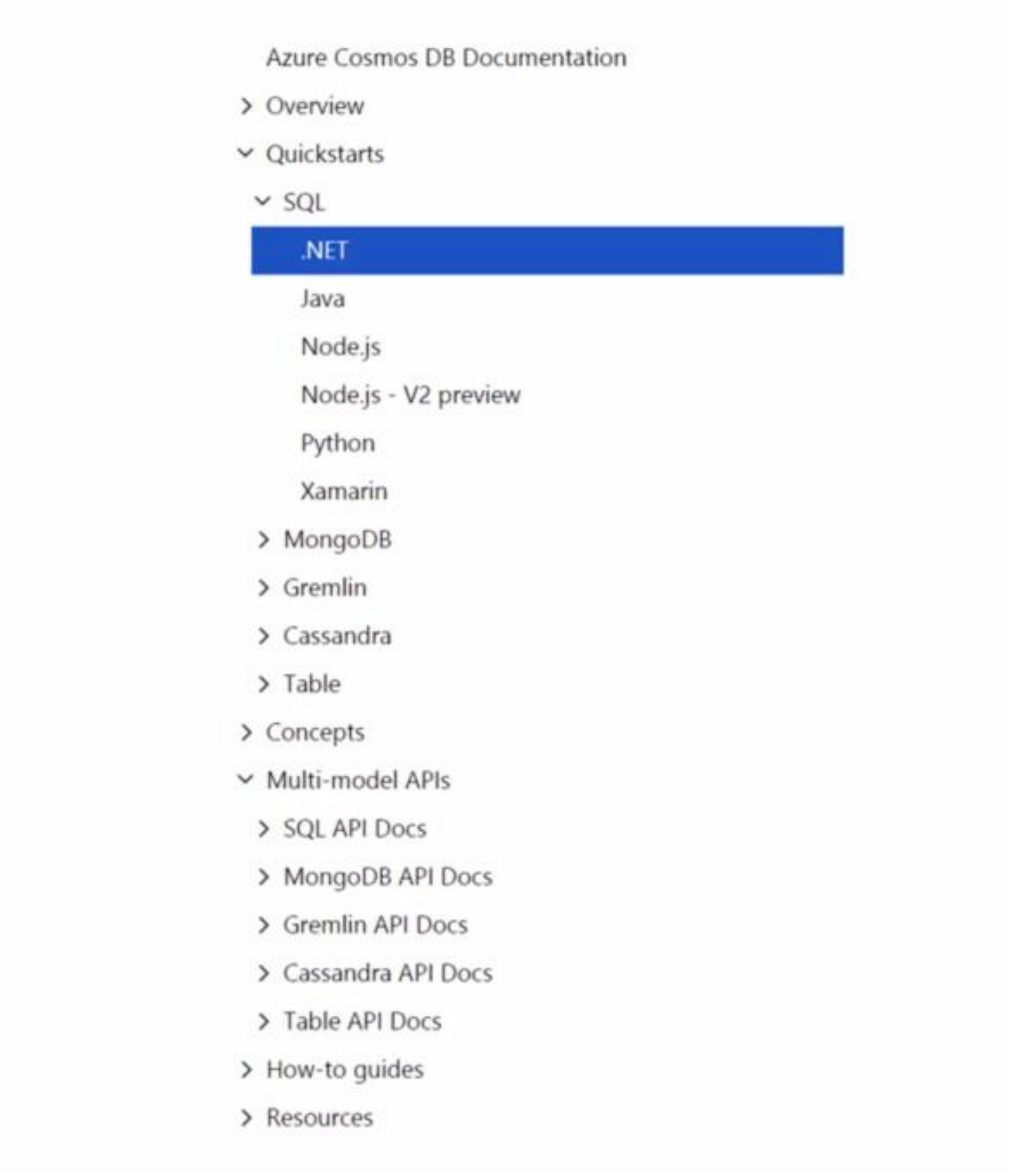
```
{
        "id": "1",
        "category": "personal",
        "name": "groceries",
        "description": "Pick up apples and strawberries."
}
```

Cosmos DB API

We've got our Cosmos DB account set up

- We can select a desired data API
- Create databases
- Write arbitrary data
- Execute queries all from the browser

We're ready to administer our data intelligently.



Azure Cosmos DB Documentation

- › Overview
- ⌄ Quickstarts
  - ⌄ SQL
    - .NET
    - Java
    - Node.js
    - Node.js - V2 preview
    - Python
    - Xamarin
  - › MongoDB
  - › Gremlin
  - › Cassandra
  - › Table
- › Concepts
- ⌄ Multi-model APIs
  - › SQL API Docs
  - › MongoDB API Docs
  - › Gremlin API Docs
  - › Cassandra API Docs
  - › Table API Docs
- › How-to guides
- › Resources

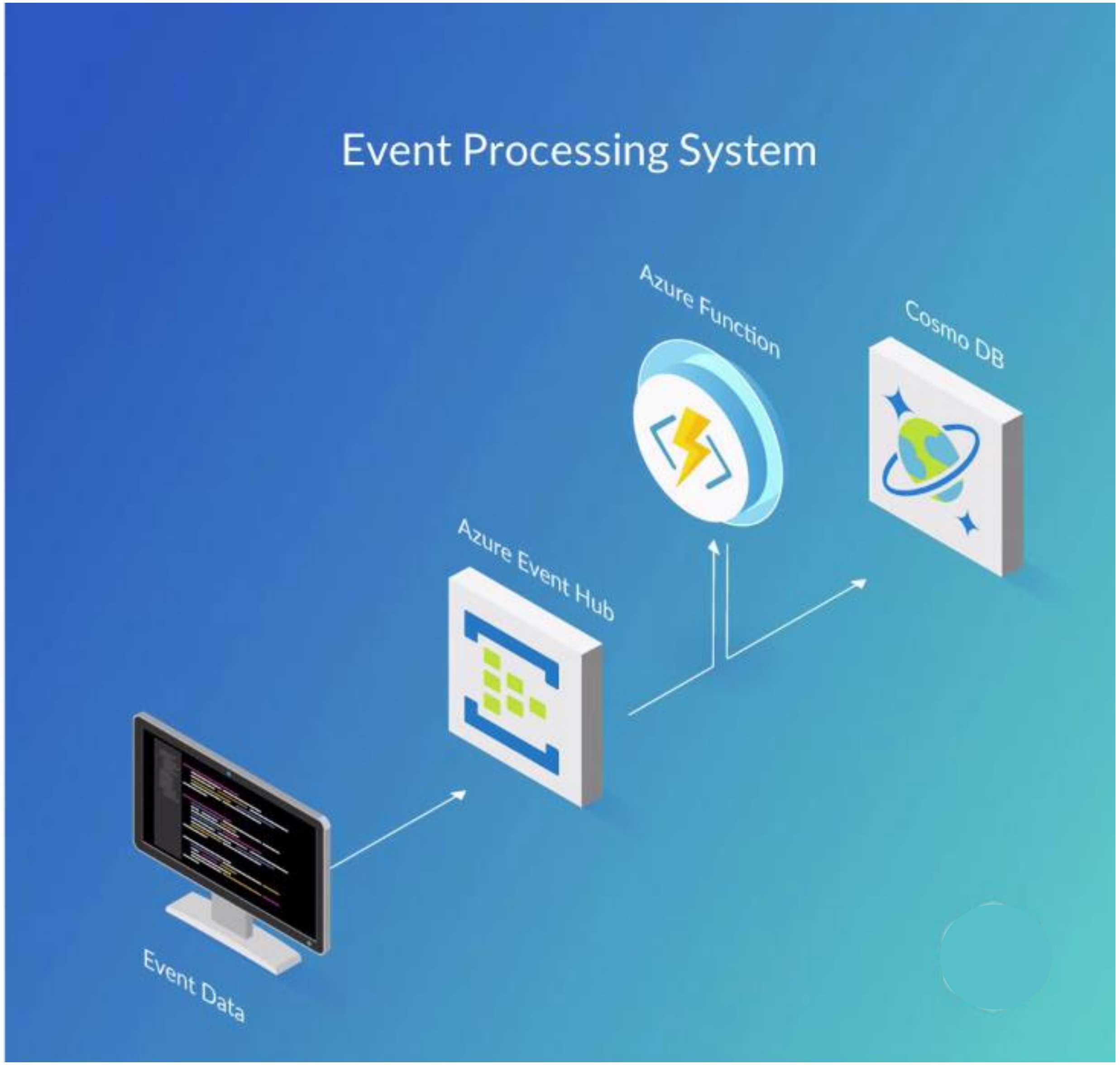
Introduction to Creating an App with Cosmos DB

By now you should have a pretty strong theoretical understanding of Cosmos DB

- We walked through its feature set
- Explained its API
- Demonstrated its tooling
- Covered how the technology can serve as a backend for a variety of use cases

Event Processing System

To keep things simple, we are not going to create a frontend. The app we create will be a simple event processing system. It will take in event data using Azure Event Hub, transmit the events to an Azure Function where we can perform transformations, and then finally save everything to Cosmos DB.



SQL Database

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

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

vCores

vCore gives you greater control over what compute and storage resources you create and pay for. vCore model enables you to configure resources independently

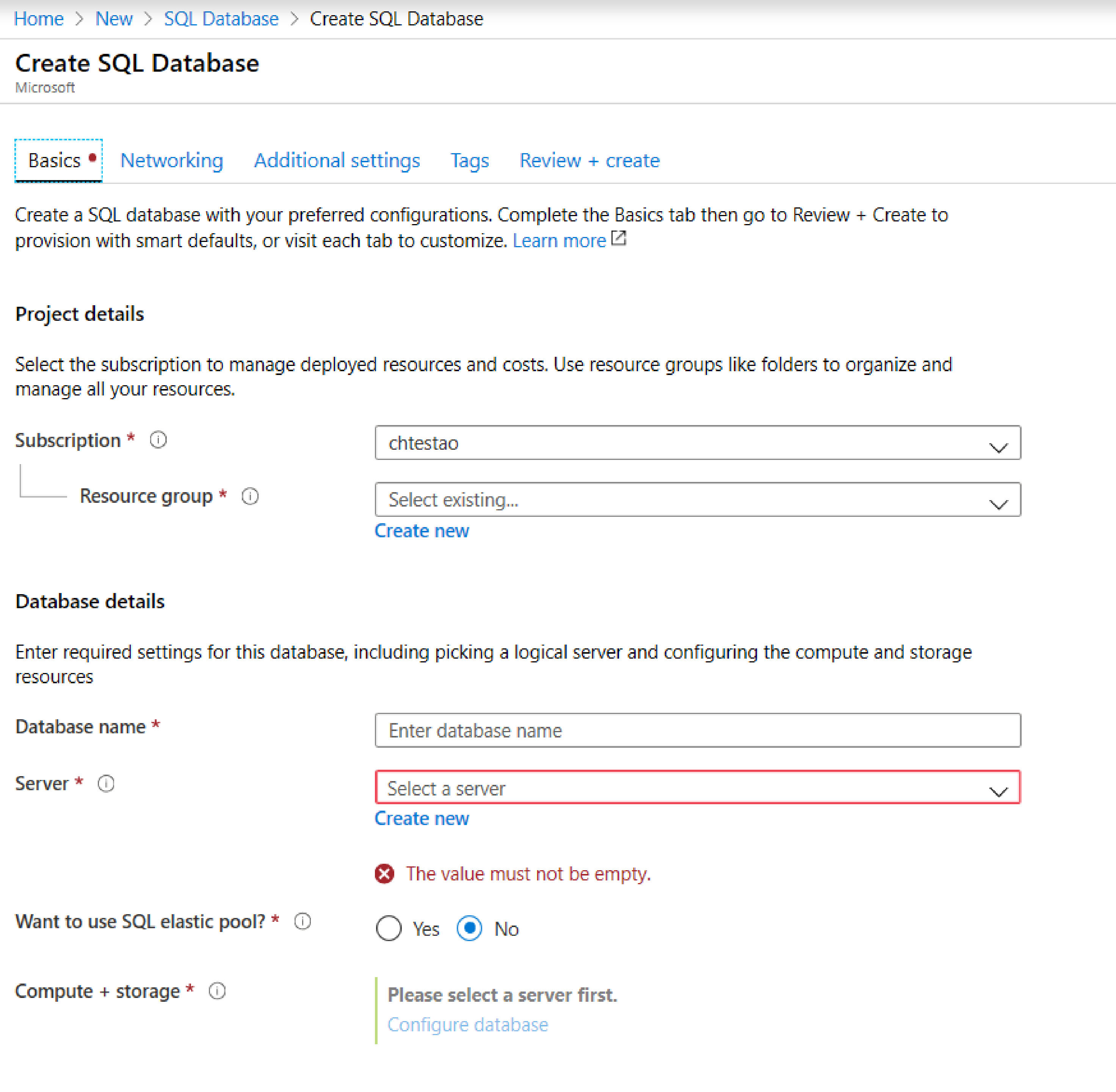
SQL elastic pools

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

SQL Managed Instances

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

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

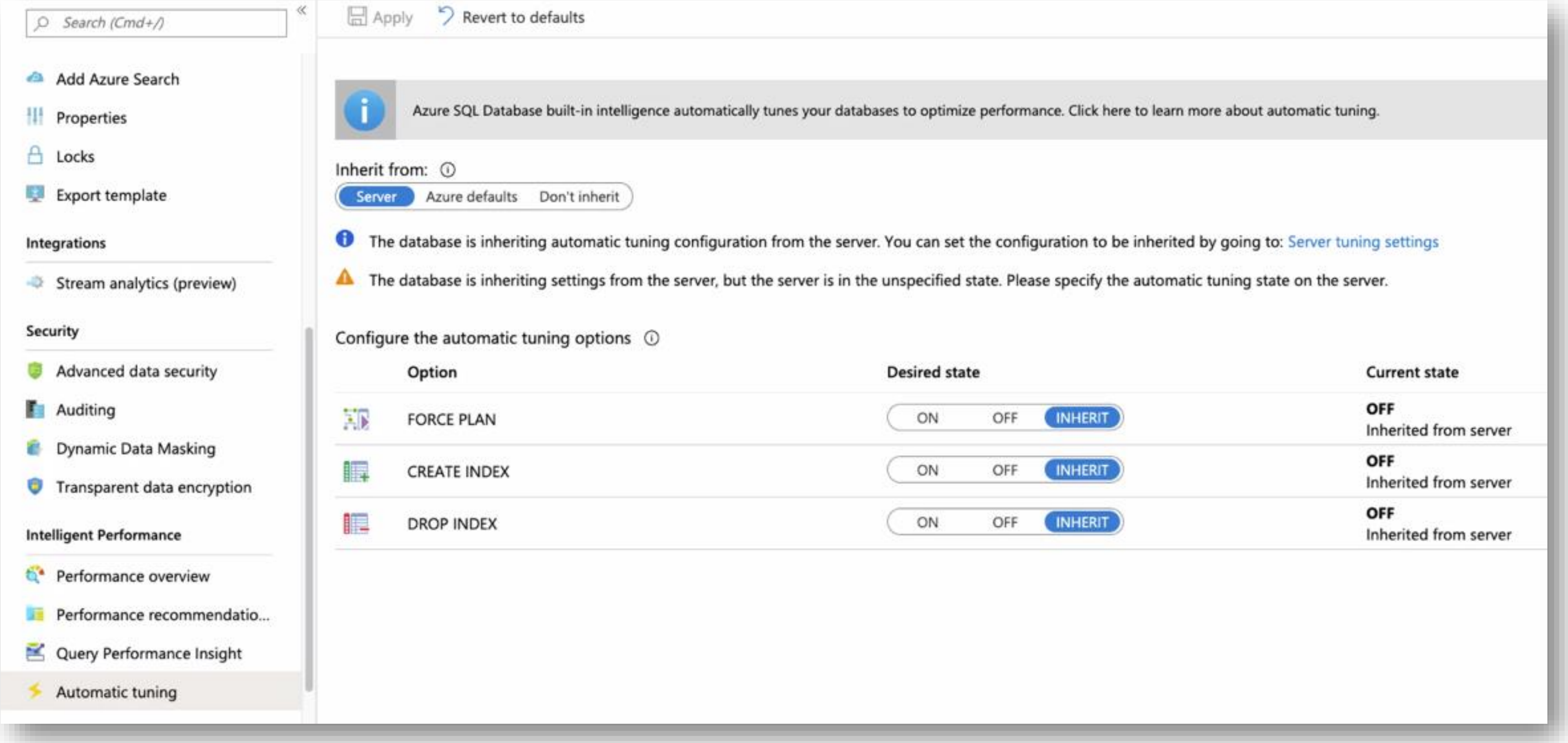
Use Azure Cloud Shell to connect to your Azure SQL database

```
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
```



Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to learn more about automatic tuning.

Inherit from: [✓ Server] Azure defaults Don't inherit

The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inherited by going to: Server tuning settings

The database is inheriting settings from the server, but the server is in the unspecified state. Please specify the automatic tuning state on the server.

Configure the automatic tuning options

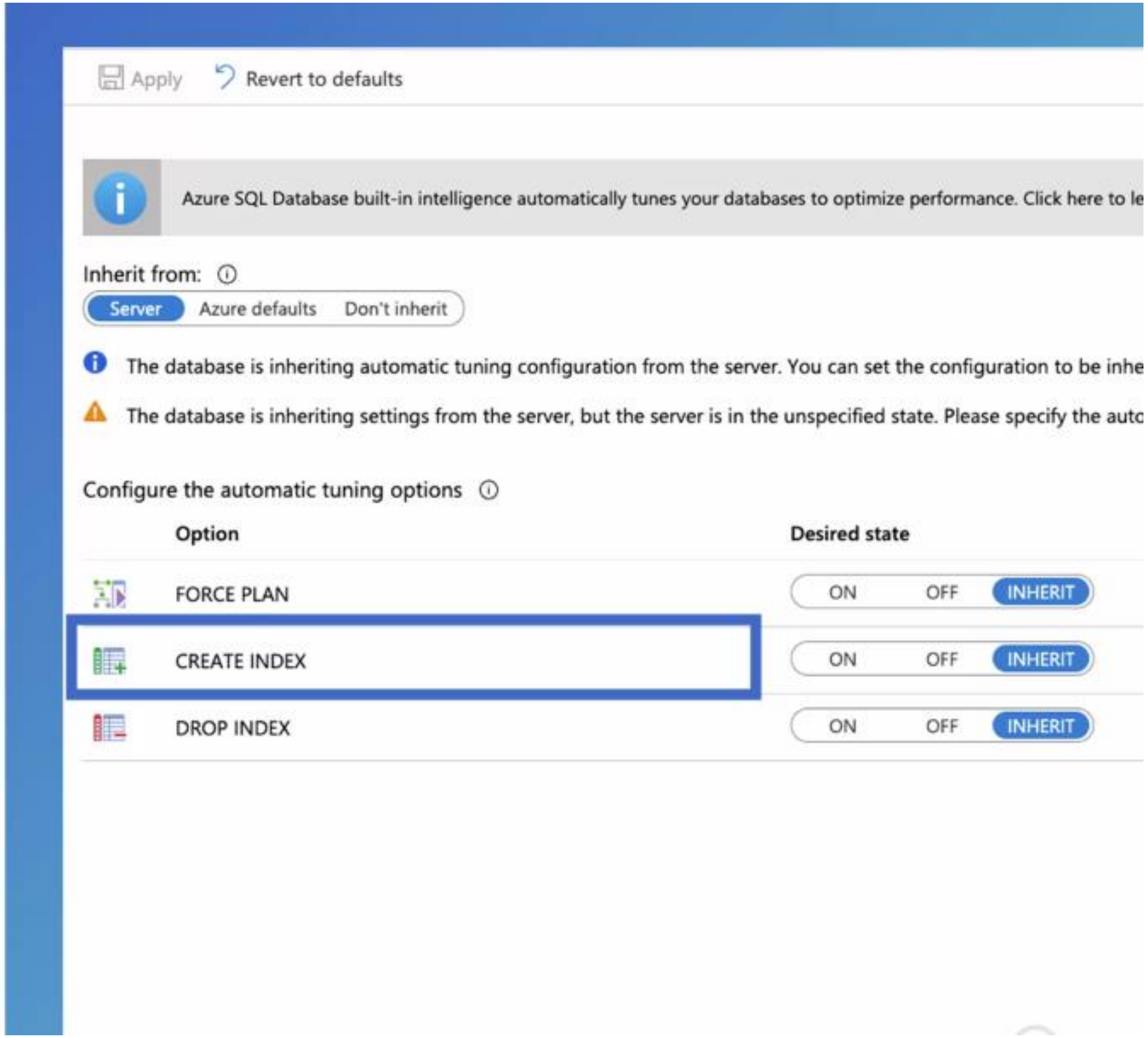
Option	Desired state	Current state
FORCE PLAN	ON OFF [✓ INHERIT]	**OFF** Inherited from server
CREATE INDEX	ON OFF [✓ INHERIT]	**OFF** Inherited from server
DROP INDEX	ON OFF [✓ INHERIT]	**OFF** Inherited from server

CREATE INDEX

This option will determine whether queries would perform better if particular indexes were created

Then it actually creates those indexes automatically!

It even monitors the performance of queries afterward to make sure that the new indexes really have improved performance



Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to le

Inherit from: [✓ Server] Azure defaults Don't inherit

The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inhe

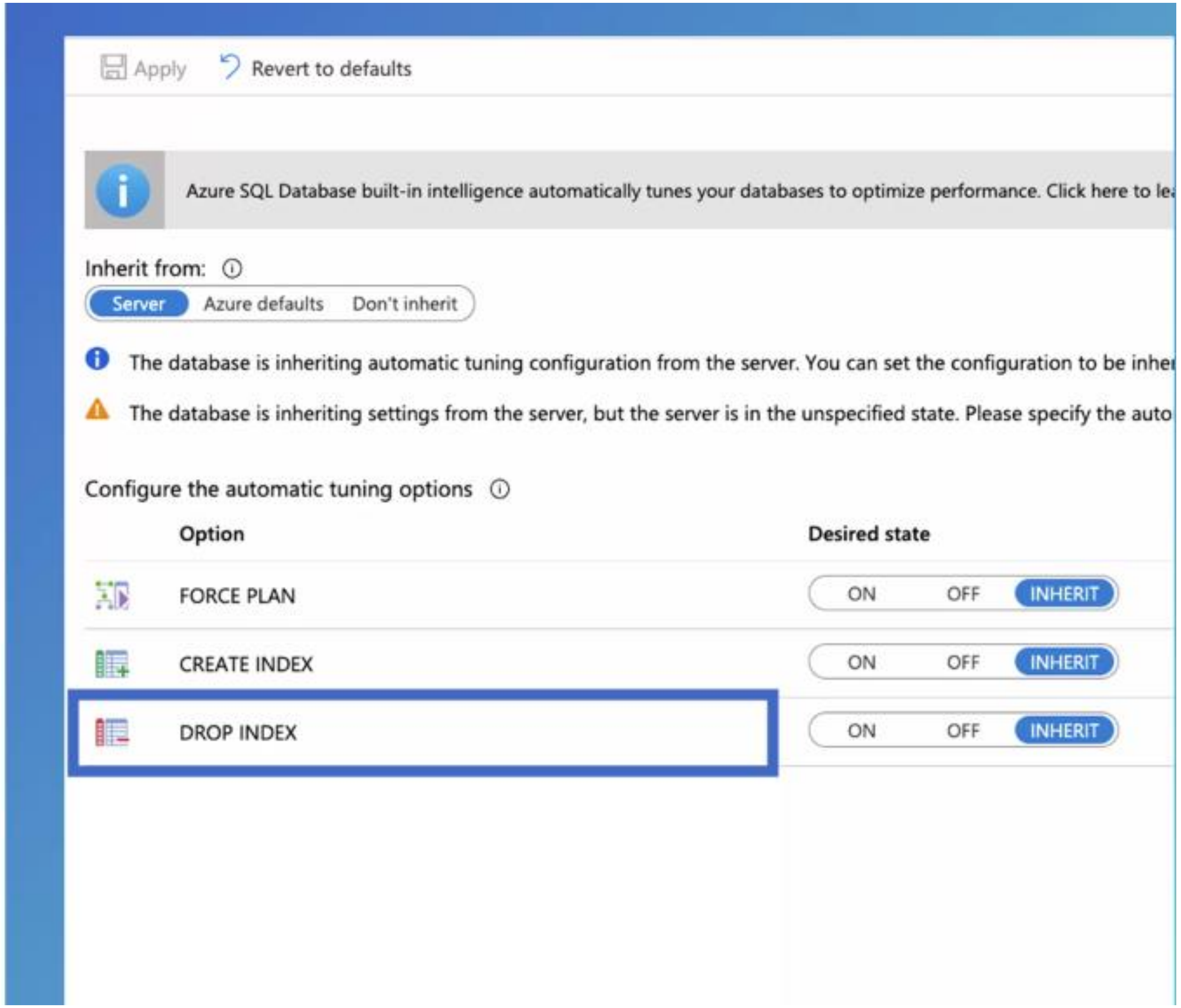
The database is inheriting settings from the server, but the server is in the unspecified state. Please specify the auto

Configure the automatic tuning options

Option	Desired state
FORCE PLAN	ON OFF [✓ INHERIT]
CREATE INDEX	ON OFF [✓ INHERIT]
DROP INDEX	ON OFF [✓ INHERIT]

DROP INDEX

Find indexes that can be safely removed without impacting performance



Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to le

Inherit from: [✓ Server] Azure defaults Don't inherit

The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inhe

The database is inheriting settings from the server, but the server is in the unspecified state. Please specify the auto

Configure the automatic tuning options

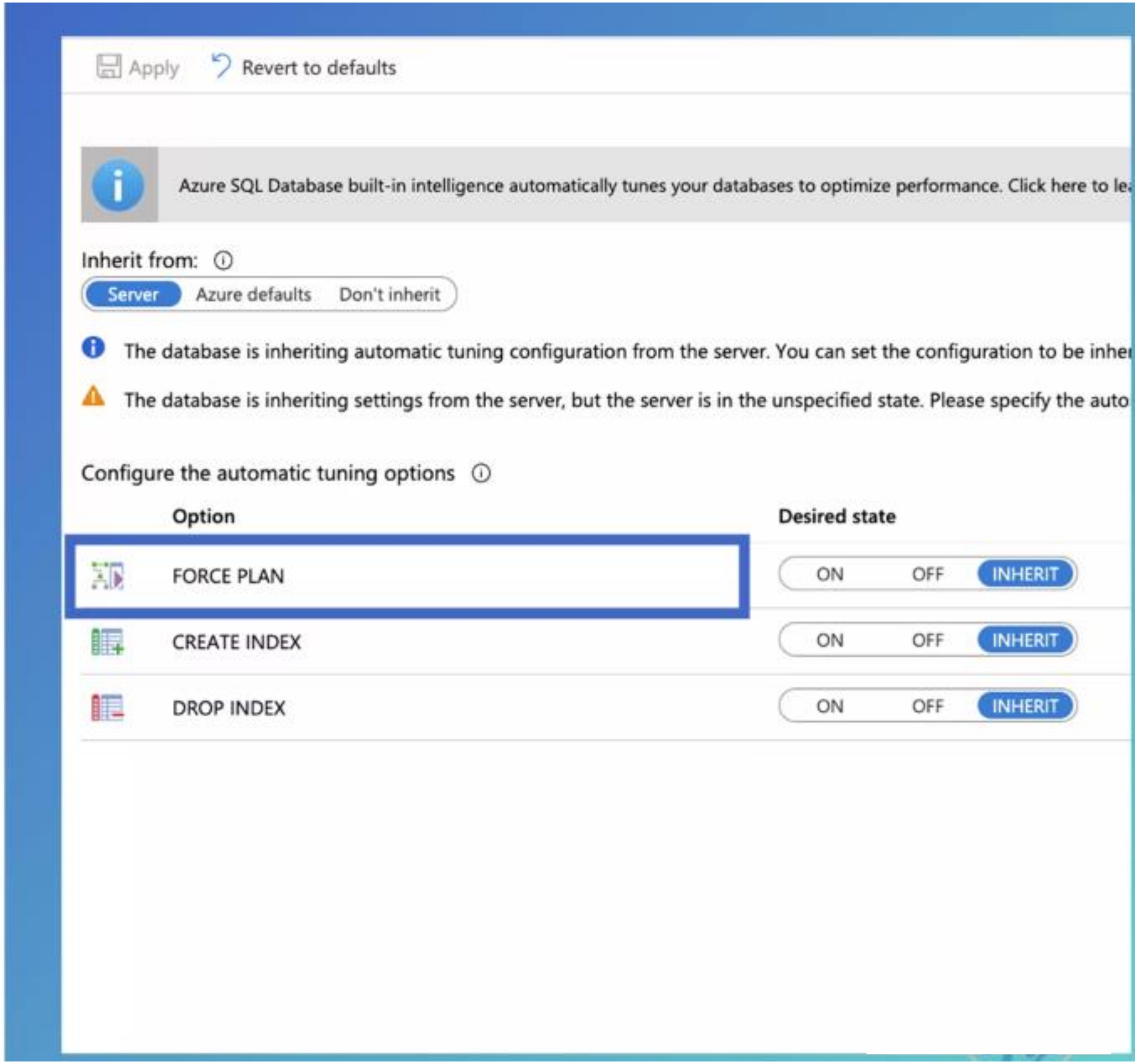
Option	Desired state
FORCE PLAN	ON OFF [✓ INHERIT]
CREATE INDEX	ON OFF [✓ INHERIT]
DROP INDEX	ON OFF [✓ INHERIT]

FORCE PLAN

When you run a query, the database engine can execute that query in a variety of different ways, so it has to decide which specific sequence of steps to use

This sequence of steps is called an execution plan

If you enable FORCE PLAN, then it will monitor workloads to see which plan has the best performance for a particular query, and it will force that plan to be used



Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to le

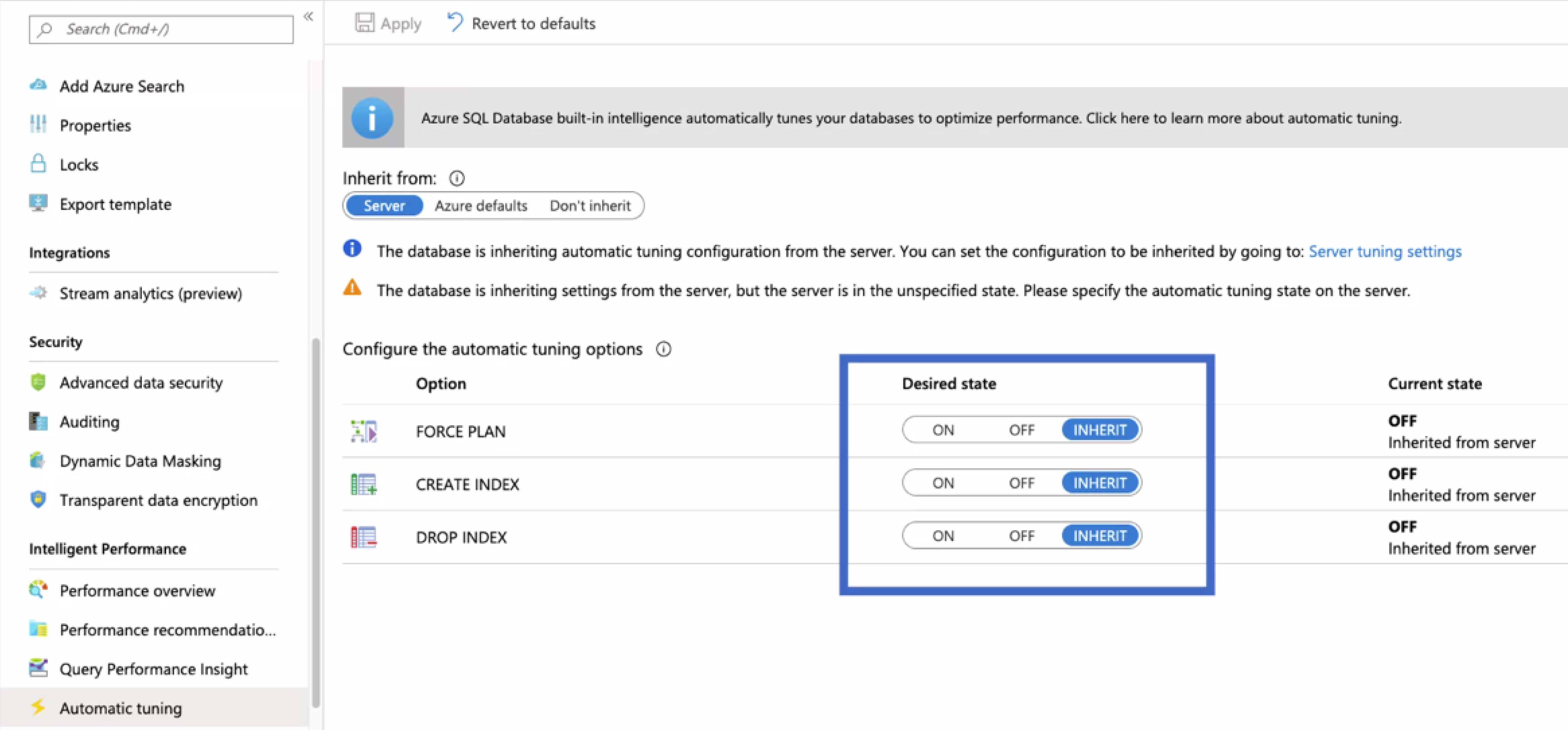
Inherit from: [✓ Server] Azure defaults Don't inherit

The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inhe

The database is inheriting settings from the server, but the server is in the unspecified state. Please specify the auto

Configure the automatic tuning options

Option	Desired state
FORCE PLAN	ON OFF [✓ INHERIT]
CREATE INDEX	ON OFF [✓ INHERIT]
DROP INDEX	ON OFF [✓ INHERIT]



Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to learn more about automatic tuning.

Inherit from: [✓ Server] Azure defaults Don't inherit

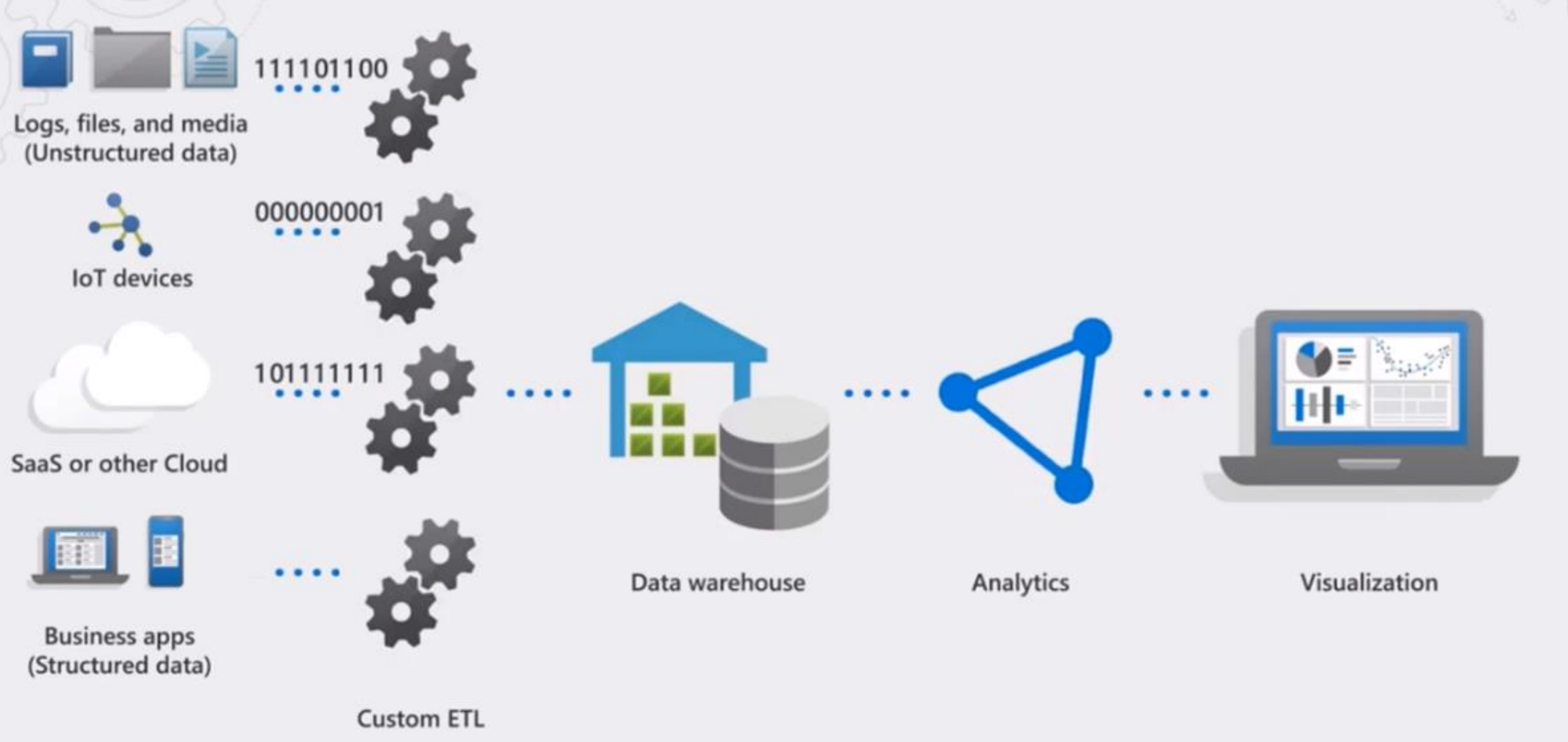
The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inherited by going to: Server tuning settings

The database is inheriting settings from the server, but the server is in the unspecified state. Please specify the automatic tuning state on the server.

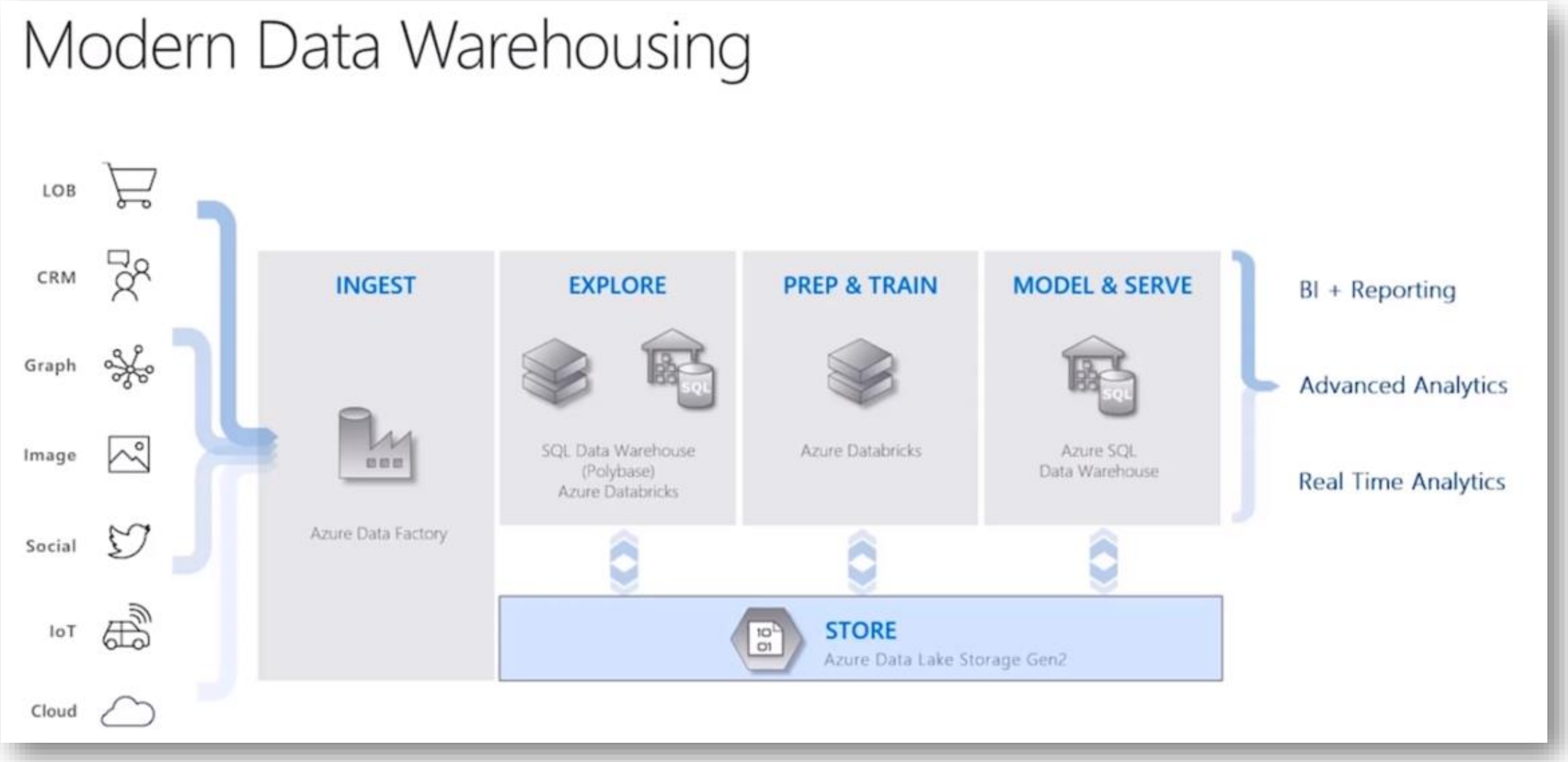
Configure the automatic tuning options

Option	Desired state	Current state
FORCE PLAN	ON OFF [✓ INHERIT]	**OFF** Inherited from server
CREATE INDEX	ON OFF [✓ INHERIT]	**OFF** Inherited from server
DROP INDEX	ON OFF [✓ INHERIT]	**OFF** Inherited from server

Synapse Analytics



Old data warehouse



Modern Data Warehousing

LOB

CRM

Graph

Image

Social

IoT

Cloud

INGEST

Azure Data Factory

EXPLORE

SQL Data Warehouse
(Polybase)
Azure Databricks

PREP & TRAIN

Azure Databricks

MODEL & SERVE

Azure SQL
Data Warehouse

STORE

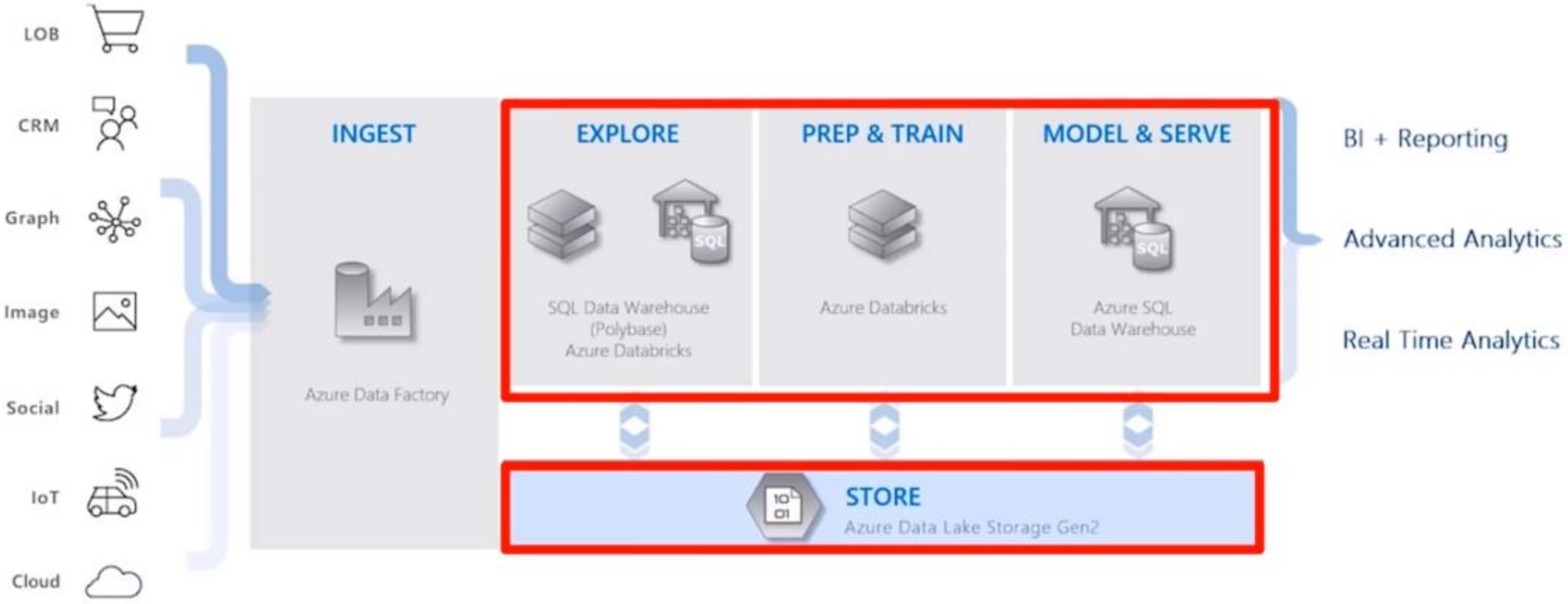
Azure Data Lake Storage Gen2

BI + Reporting

Advanced Analytics

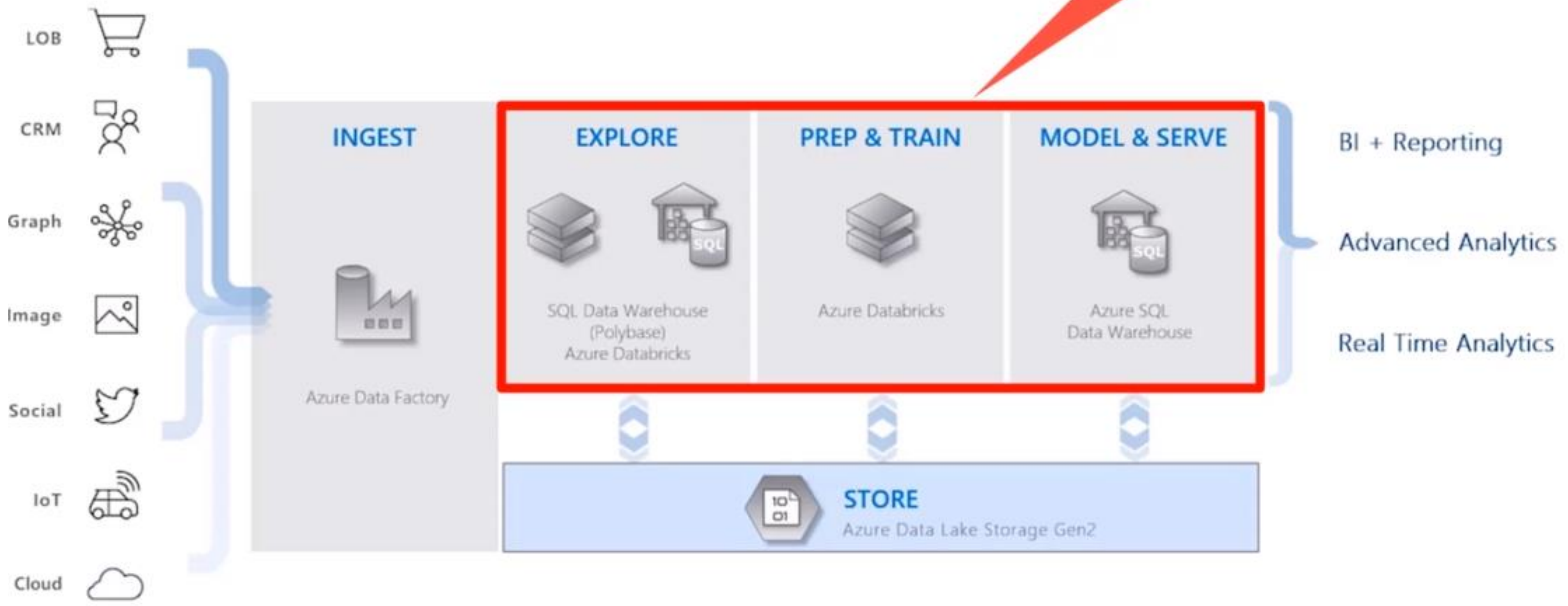
Real Time Analytics

Modern Data Warehousing



Modern Data Warehousing

DWU = CPU + Memory + IO



LOB

CRM

Graph

Image

Social

IoT

Cloud

INGEST

Azure Data Factory

EXPLORE

SQL Data Warehouse (Polybase)
Azure Databricks

PREP & TRAIN

Azure Databricks

MODEL & SERVE

Azure SQL Data Warehouse

STORE

Azure Data Lake Storage Gen2

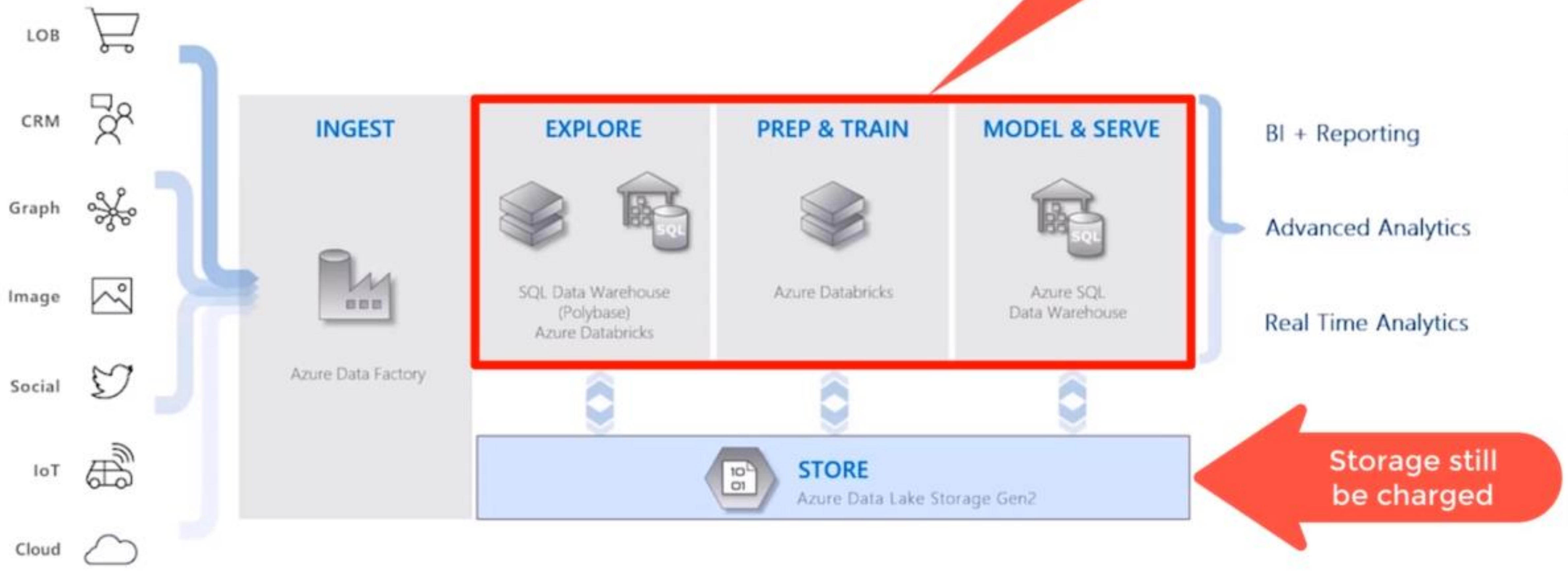
BI + Reporting

Advanced Analytics

Real Time Analytics

Modern Data Warehousing

When not in use, compute can be paused



LOB

CRM

Graph

Image

Social

IoT

Cloud

INGEST

Azure Data Factory

EXPLORE

SQL Data Warehouse (Polybase)
Azure Databricks

PREP & TRAIN

Azure Databricks

MODEL & SERVE

Azure SQL Data Warehouse

STORE

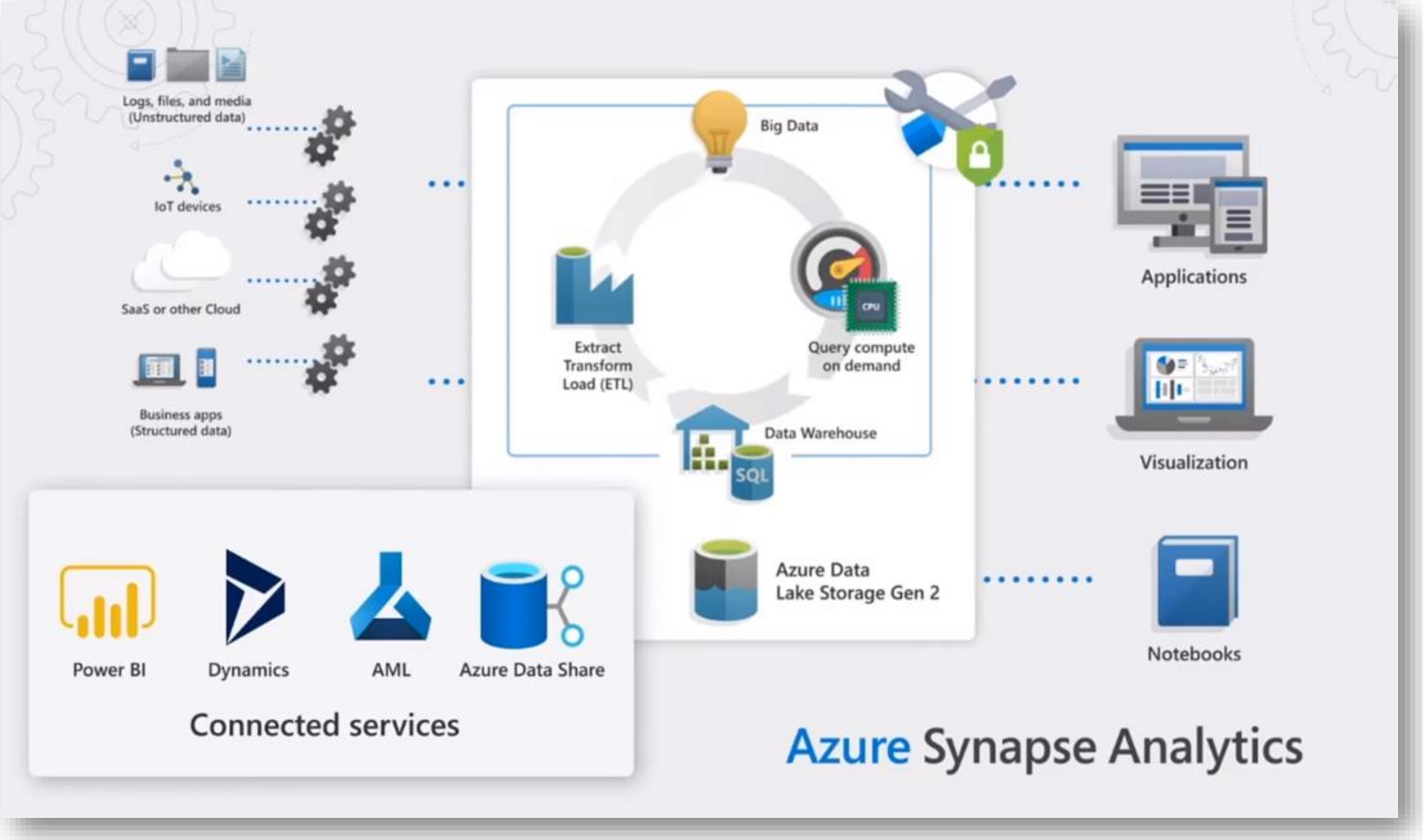
Azure Data Lake Storage Gen2

BI + Reporting

Advanced Analytics

Real Time Analytics

Storage still be charged



Logs, files, and media (Unstructured data)

IoT devices

SaaS or other Cloud

Business apps (Structured data)

Big Data

Extract Transform Load (ETL)

Query compute on demand

Data Warehouse

Azure Data Lake Storage Gen 2

Applications

Visualization

Notebooks

Power BI

Dynamics

AML

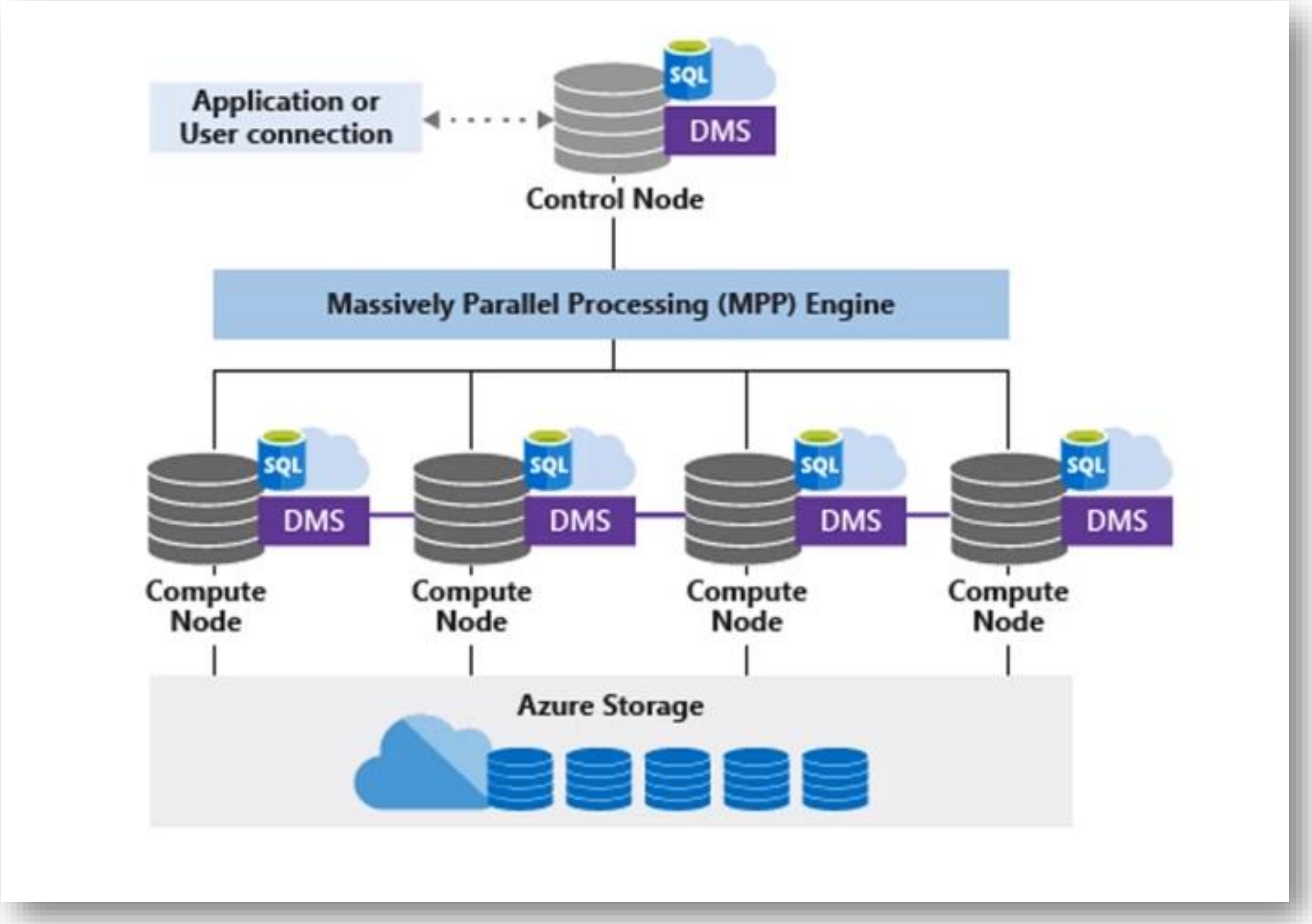
Azure Data Share

Connected services

Azure Synapse Analytics

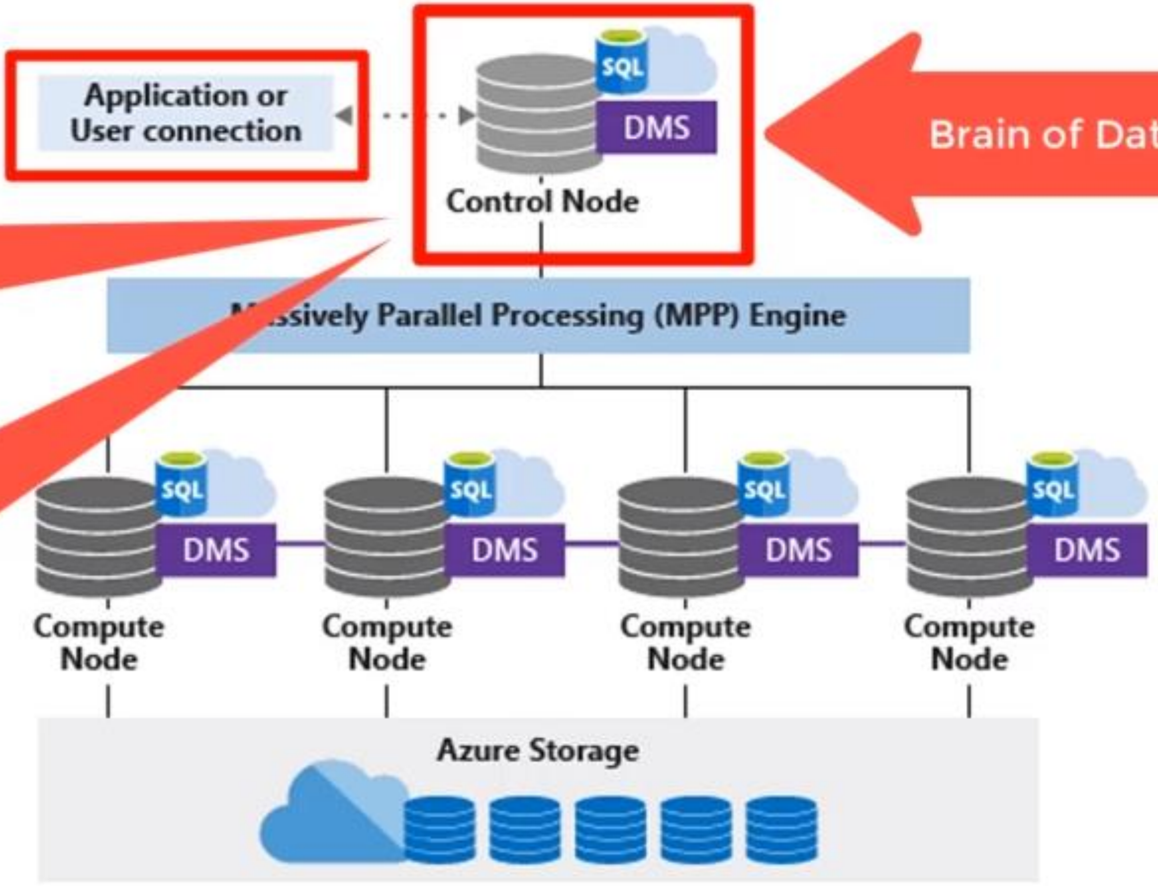
Azure Synapse Analytics

- Create Synapse SQL Pool
- Pause / Resume Compute Node
- Create Firewall Rules
- Connect with Microsoft SQL Server Management Studio


Application or User connection
SQL
DMS
Control Node
Massively Parallel Processing (MPP) Engine
SQL
DMS
SQL
DMS
SQL
DMS
SQL
DMS
Compute Node
Compute Node
Compute Node
Compute Node
Azure Storage

The front end that interacts with all applicaitons and connections.

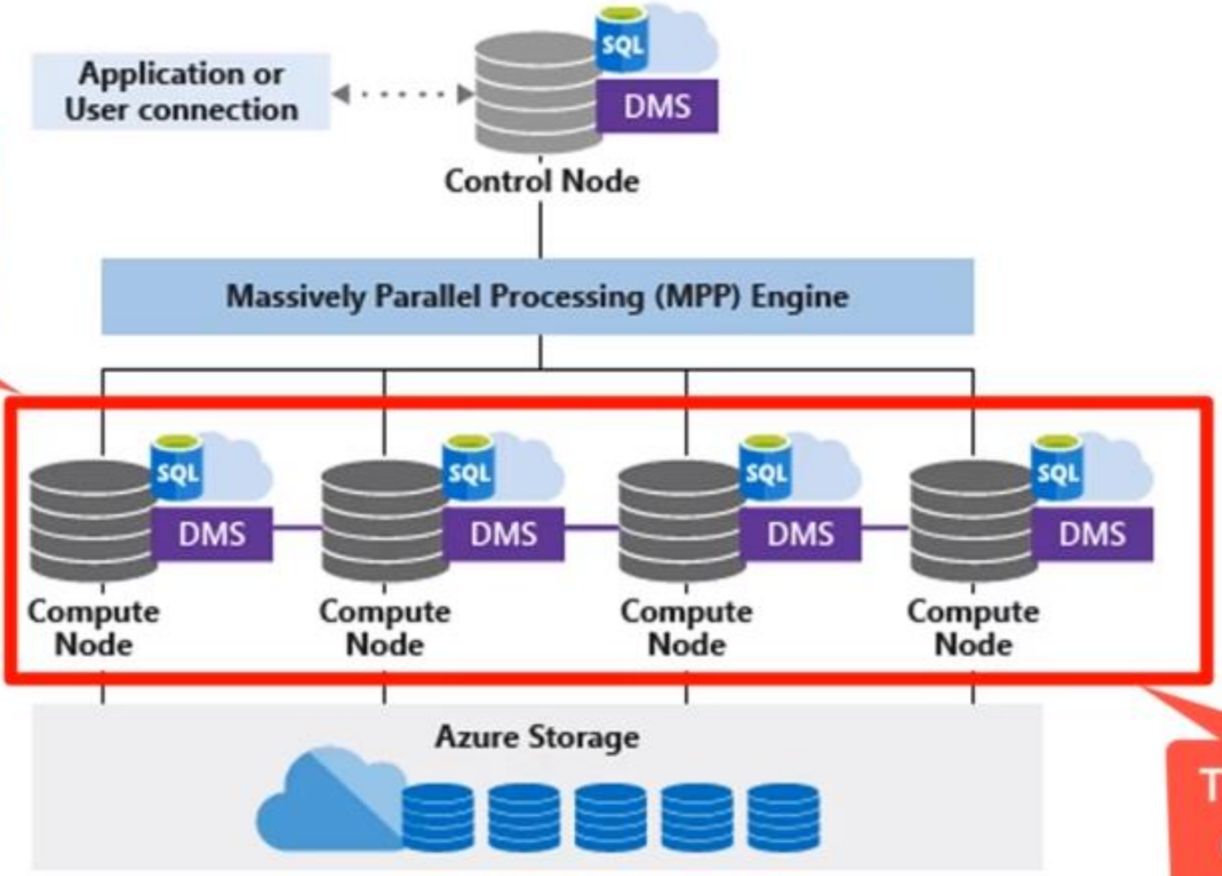
MPP engine runs on the control node to optimize and coordinate parallel queries



Brain of Data Warehouse

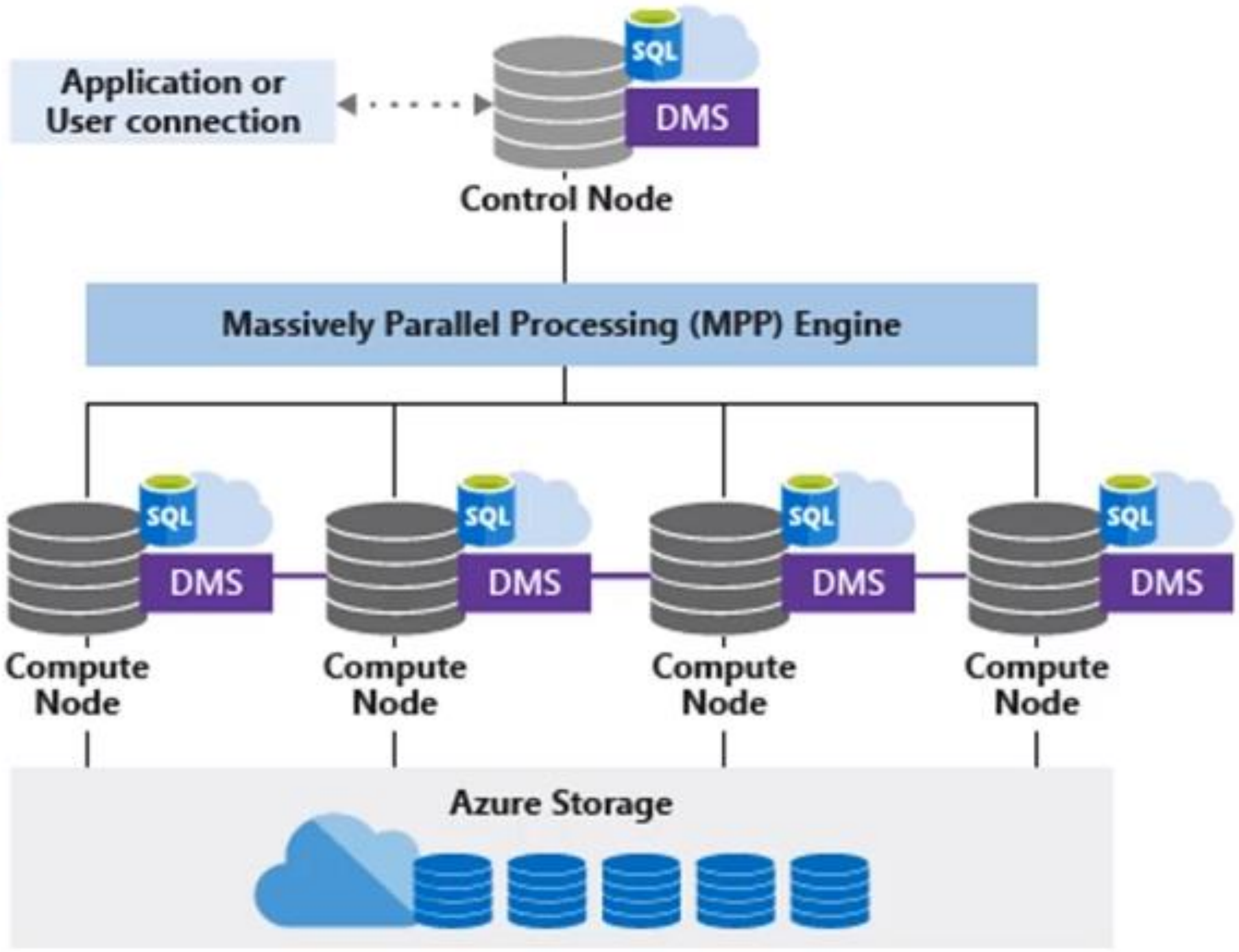
Provide the computational power for analytics.

Distribution map to compute nodes for processing.

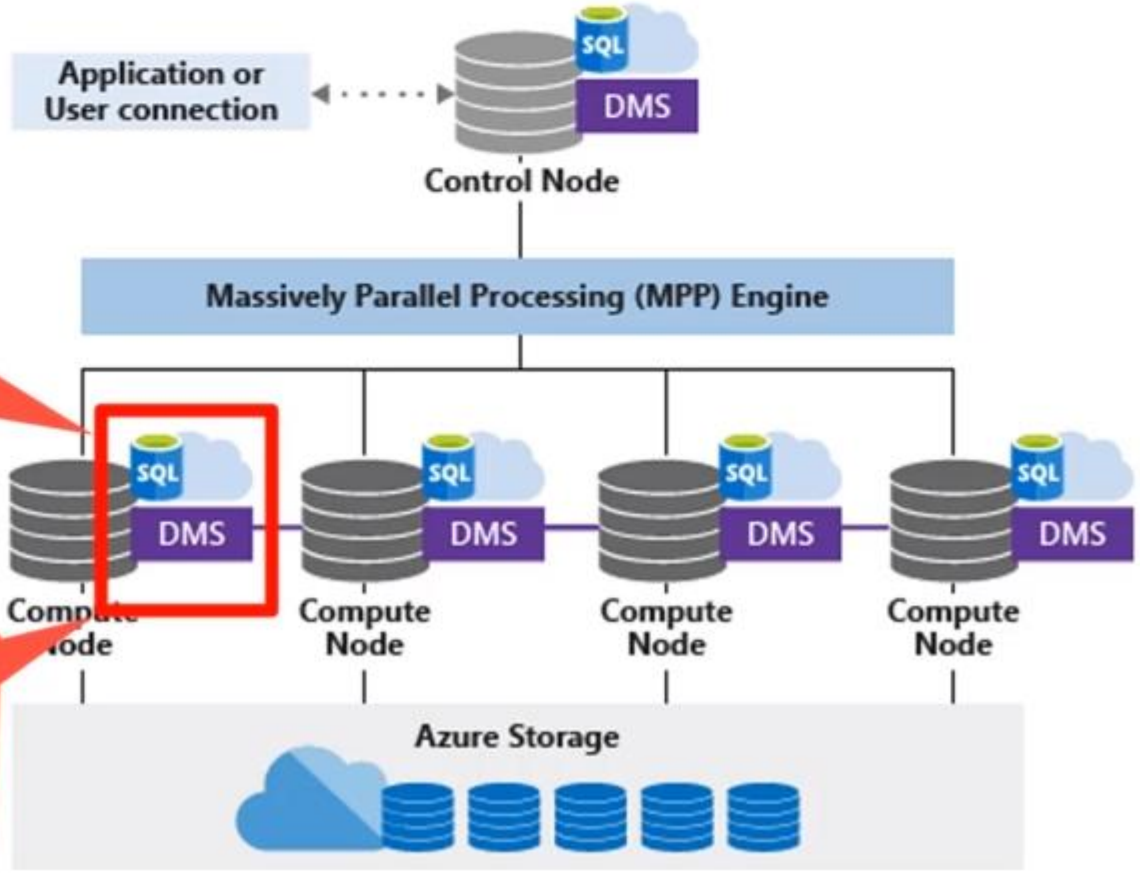


These are scaled using data warehouse unit (DWU)

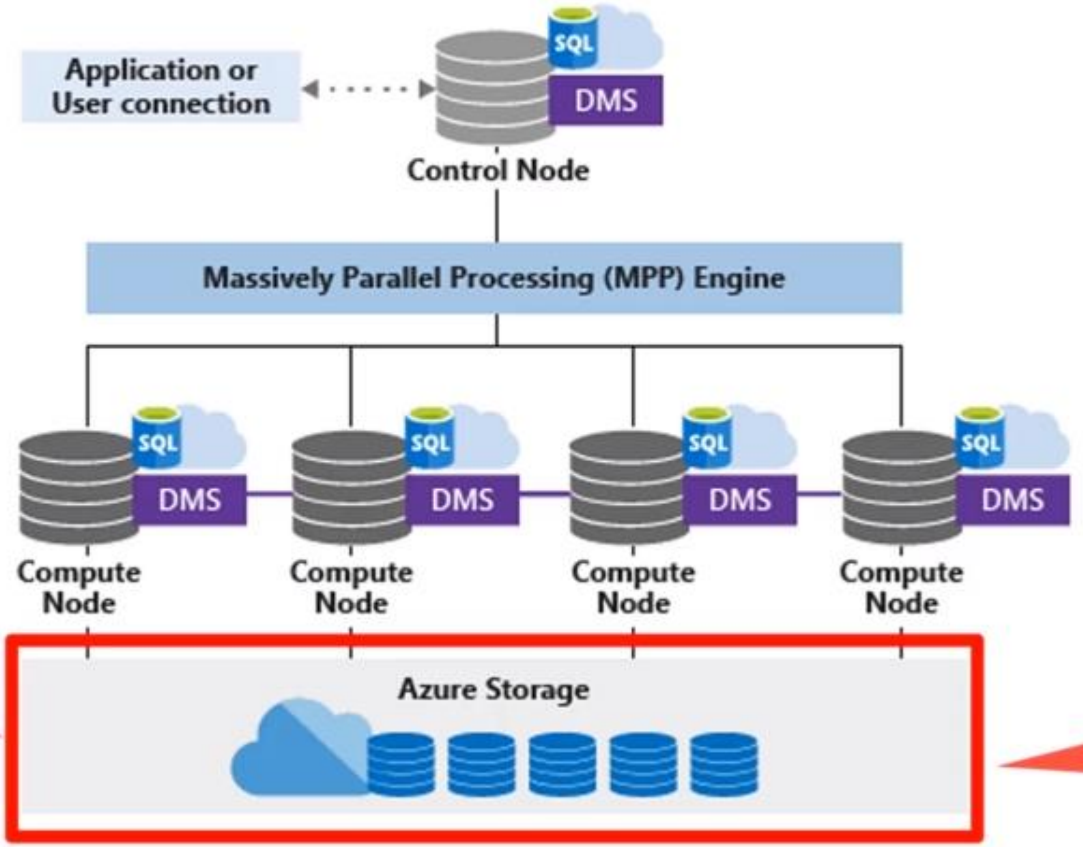
DWU	Loading 3 Tables	Ran Report
100	15	20
500	3	4



DMS is Data transport technology that coordinates data movement between compute nodes.

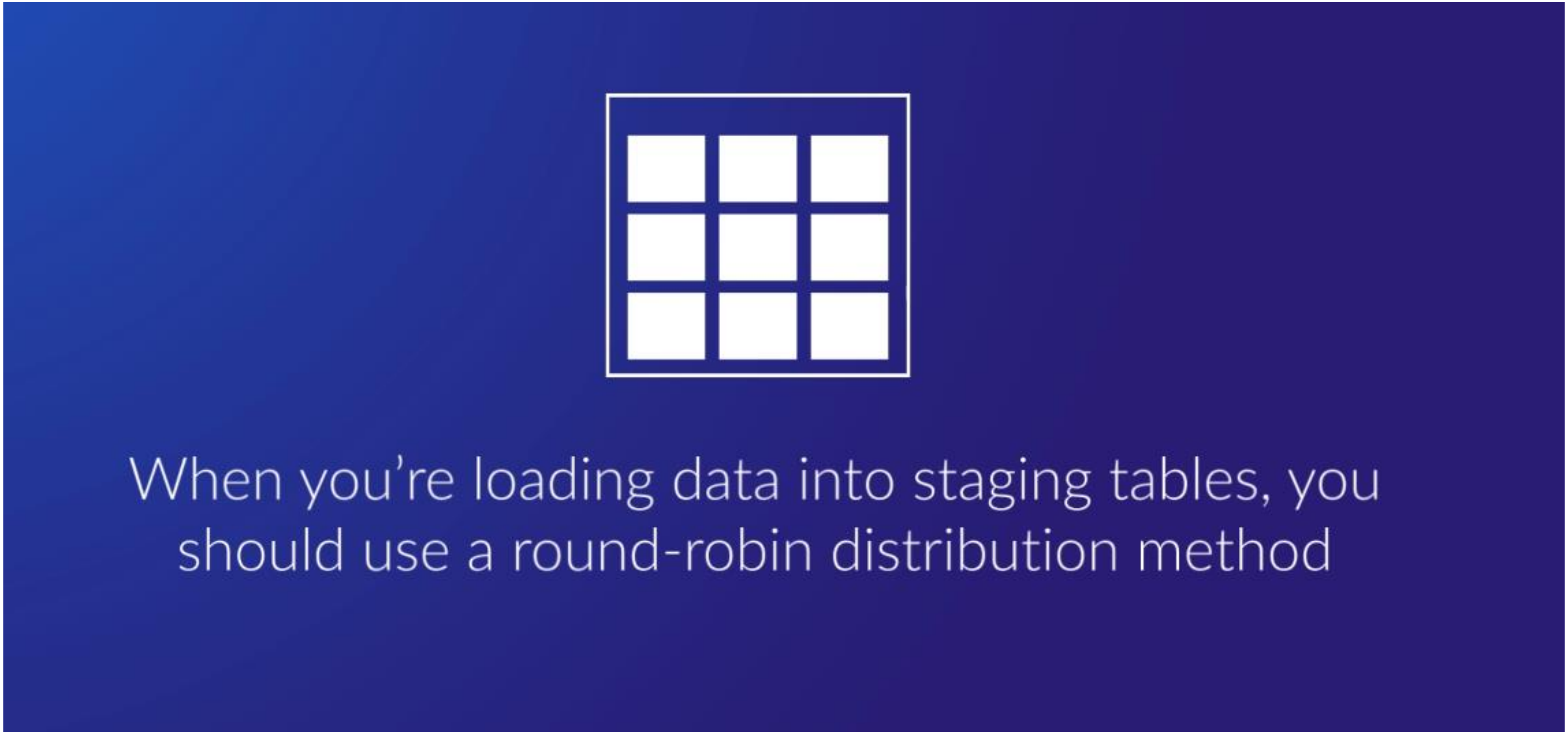


When SQL Data Warehouse runs a query, the work is divided into 60 smaller queries that run in parallel

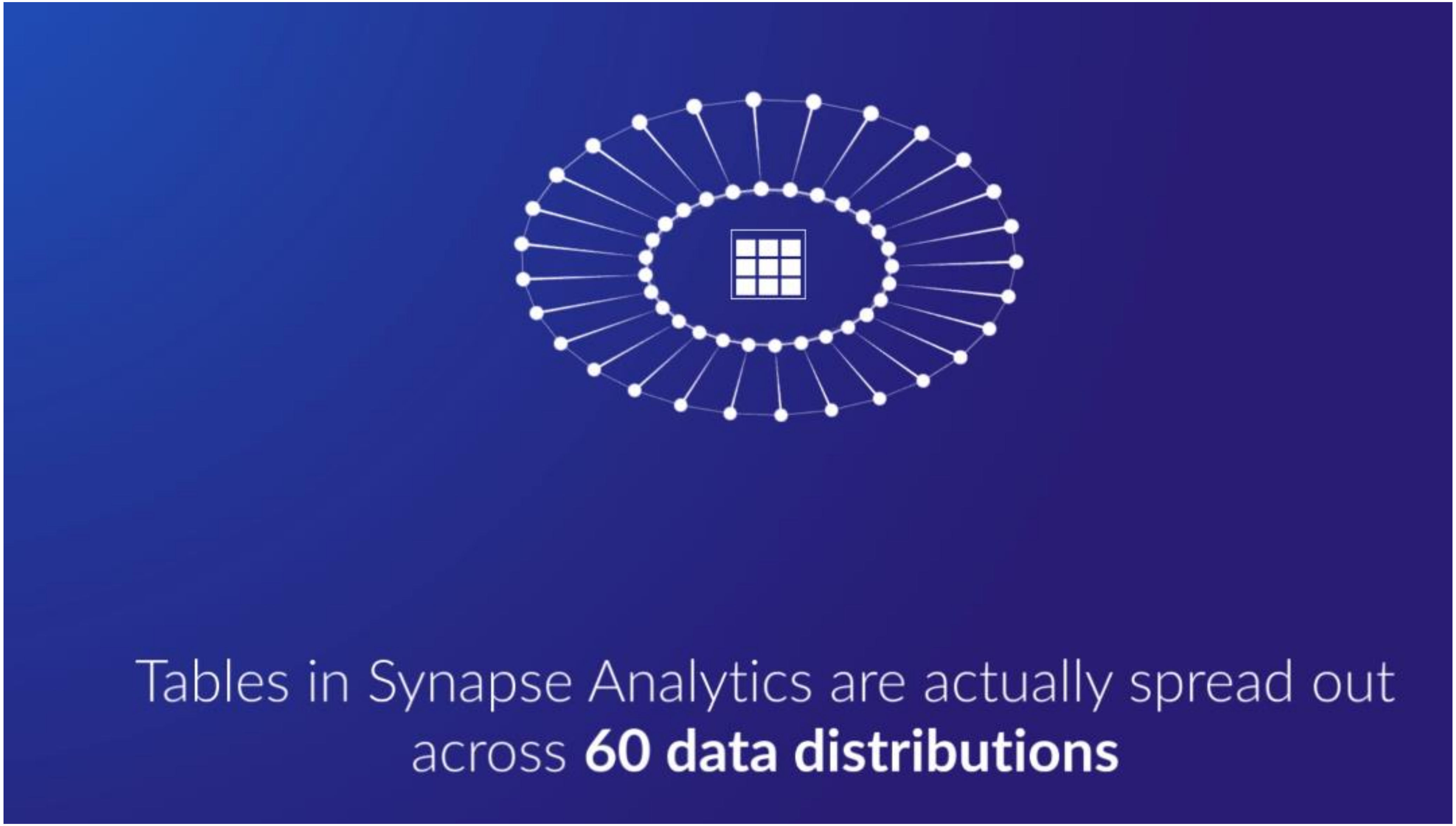


Seperate from compute in order to keep data at rest

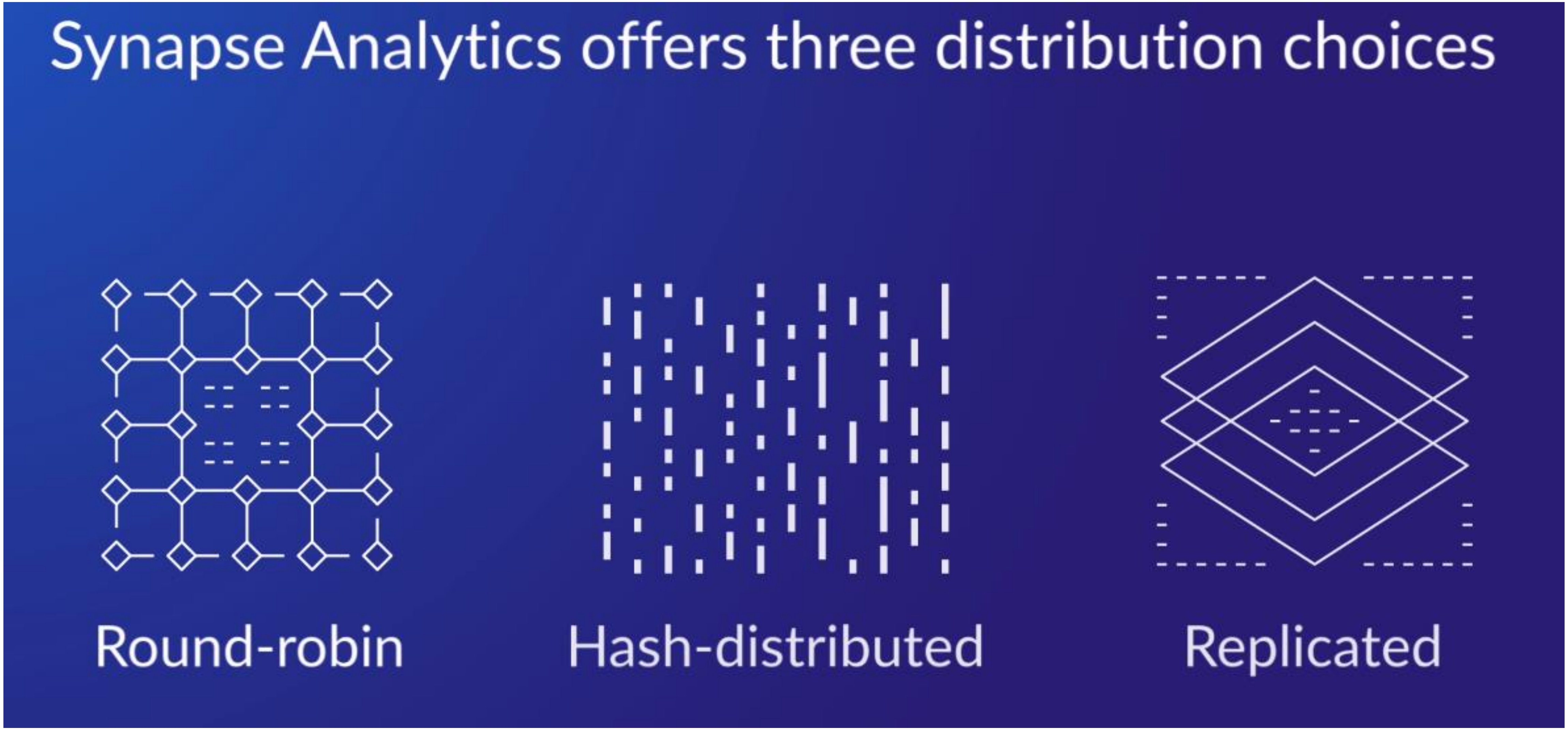
This is cheaper than data that is being analyzed.



When you're loading data into staging tables, you should use a round-robin distribution method



Tables in Synapse Analytics are actually spread out across **60 data distributions**

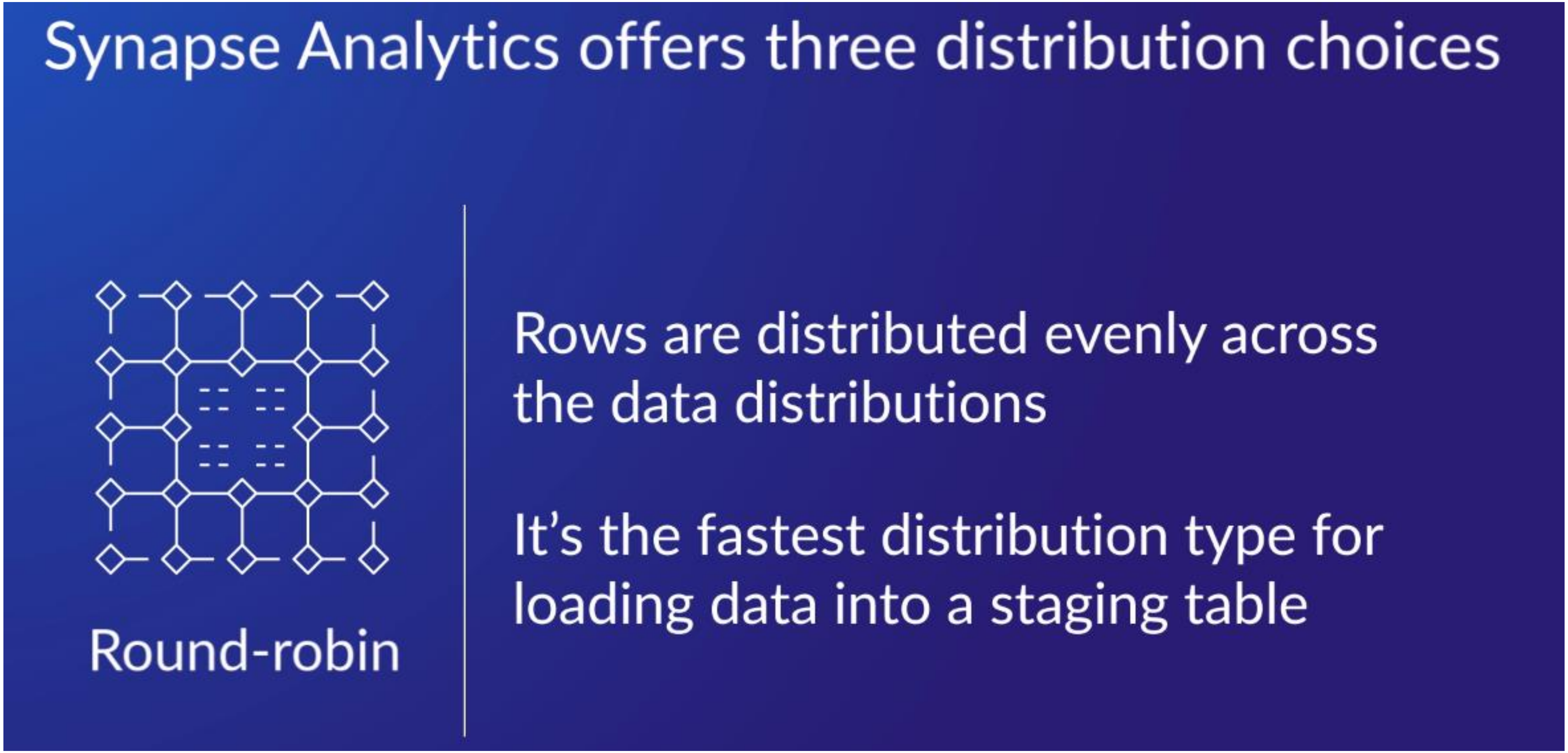


Synapse Analytics offers three distribution choices

Round-robin

Hash-distributed

Replicated

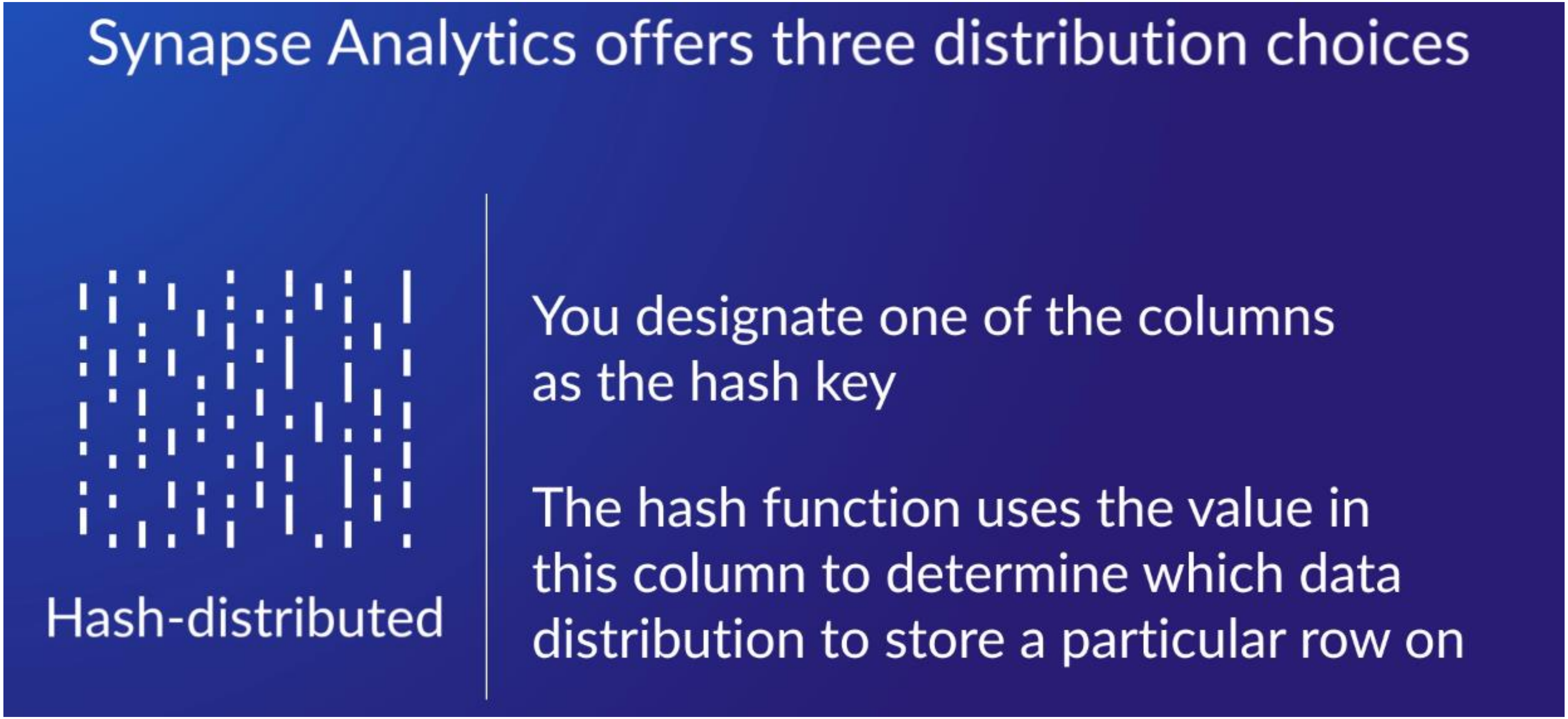


Synapse Analytics offers three distribution choices

Round-robin

Rows are distributed evenly across the data distributions

It’s the fastest distribution type for loading data into a staging table

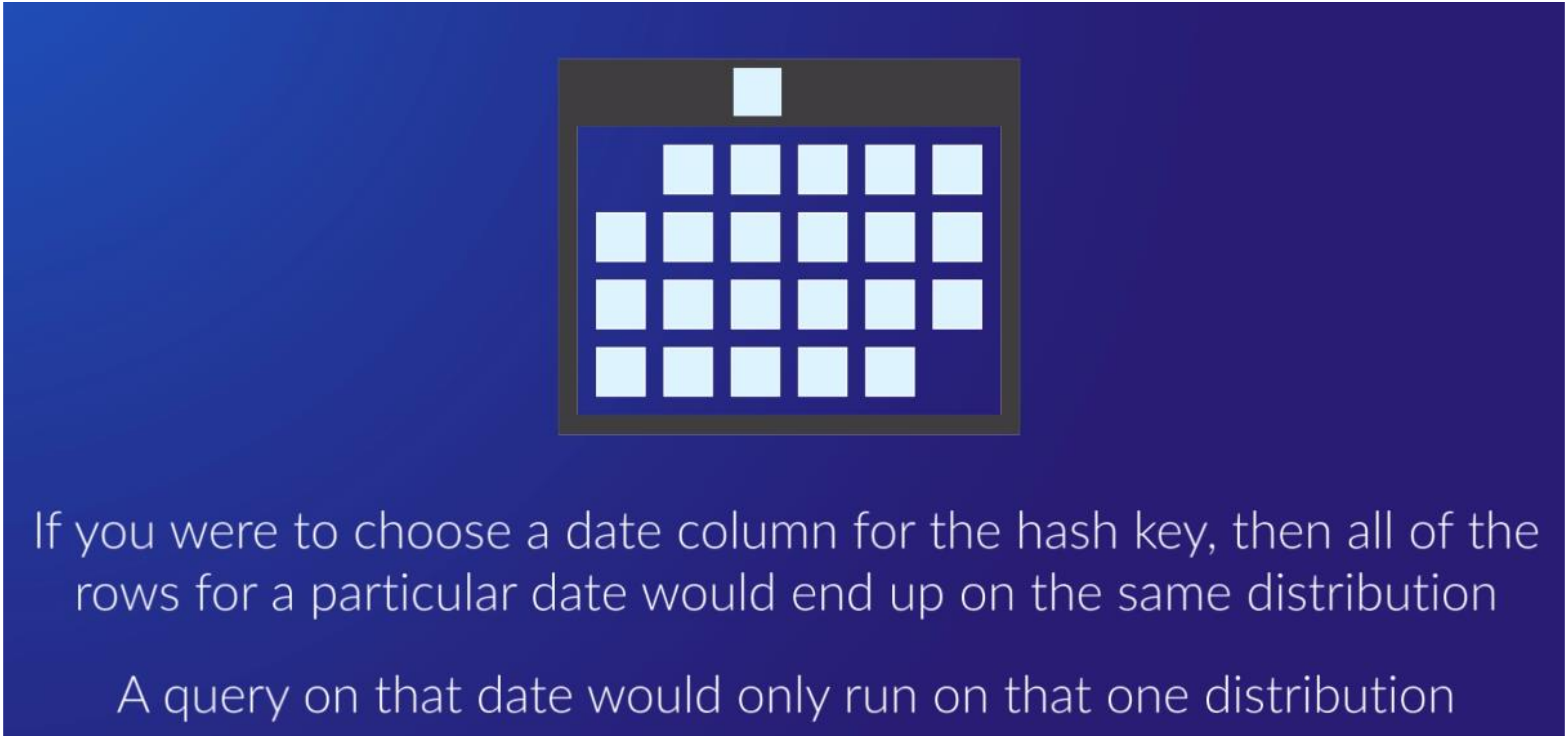


Synapse Analytics offers three distribution choices

Hash-distributed

You designate one of the columns as the hash key

The hash function uses the value in this column to determine which data distribution to store a particular row on



If you were to choose a date column for the hash key, then all of the rows for a particular date would end up on the same distribution

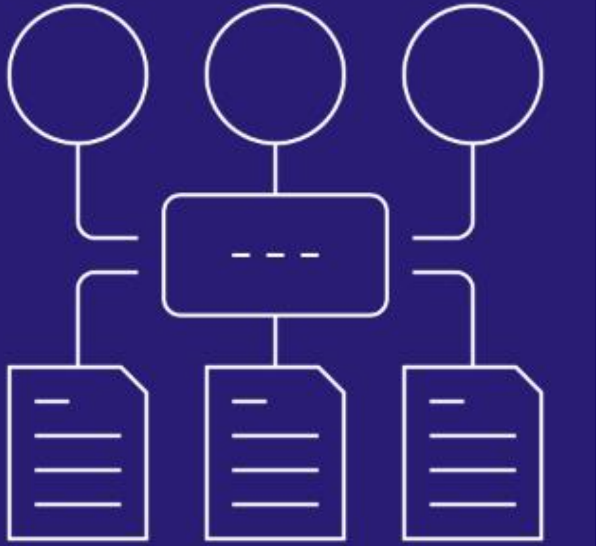
A query on that date would only run on that one distribution

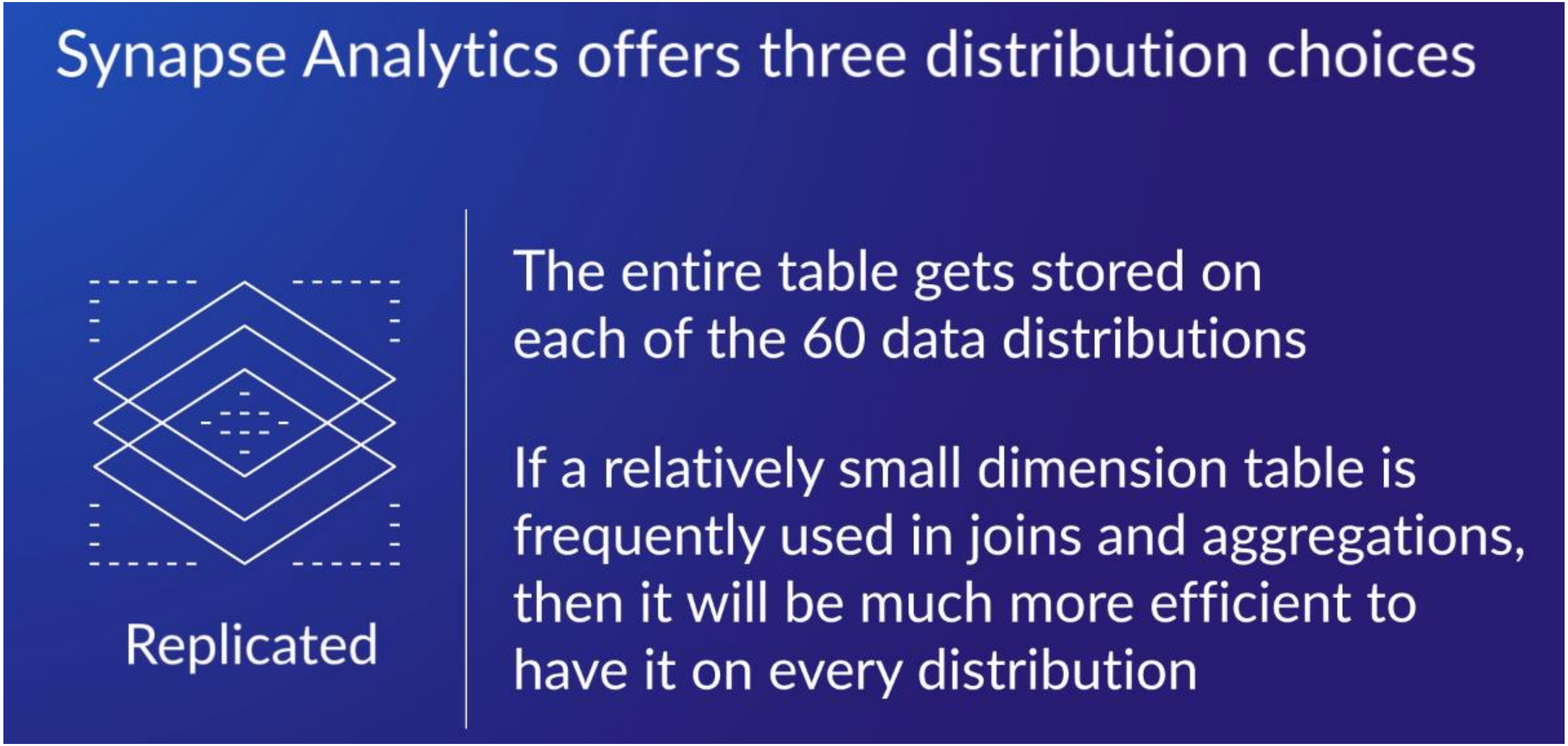
Some characteristics of a good distribution column:

1. It has many unique values so the rows will be spread out over the 60 distributions

2. It's frequently used in JOINs

3. It's not used in WHERE clauses, as this would limit query matches to only a few distributions



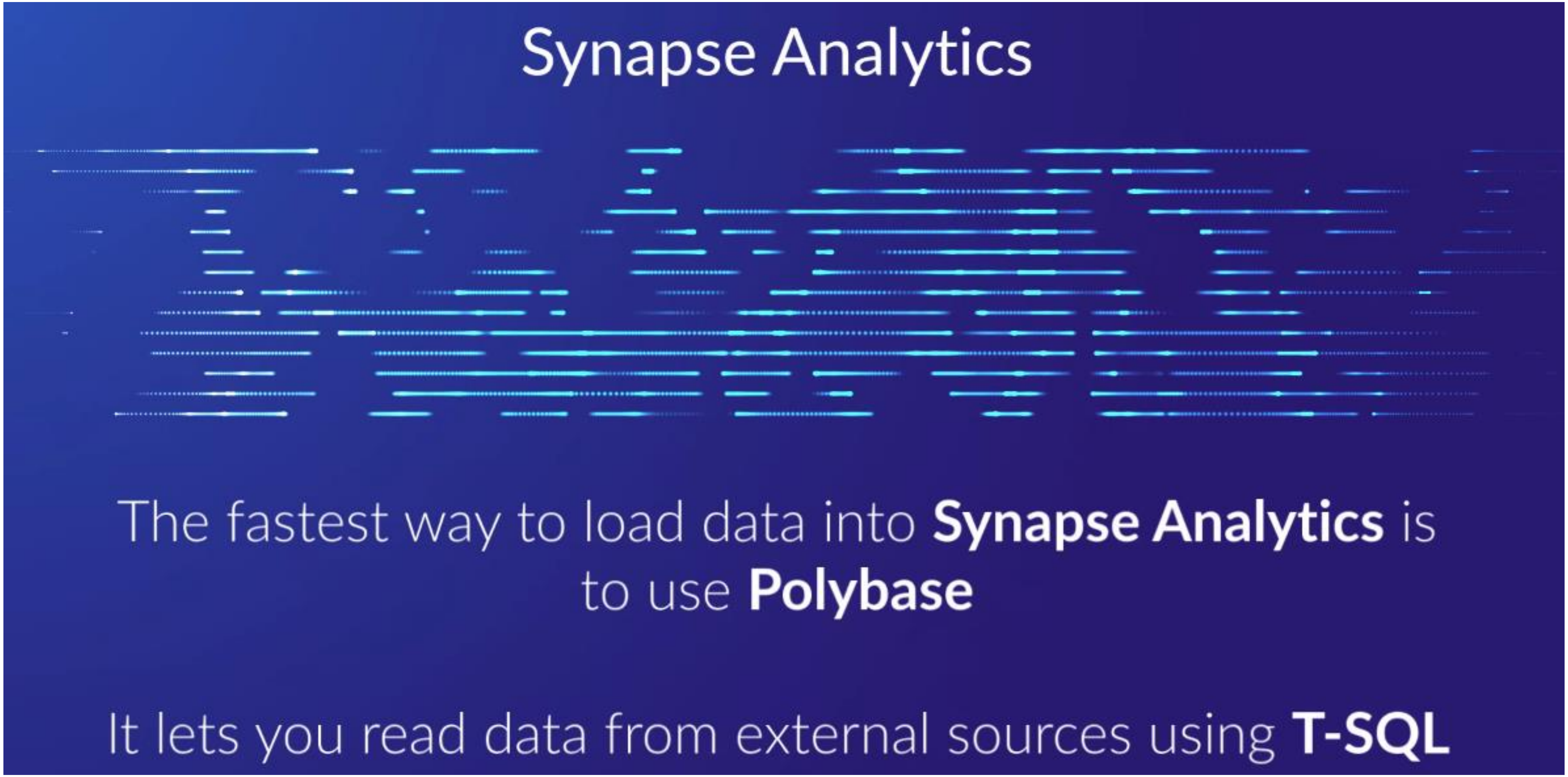


Synapse Analytics offers three distribution choices

Replicated

The entire table gets stored on each of the 60 data distributions

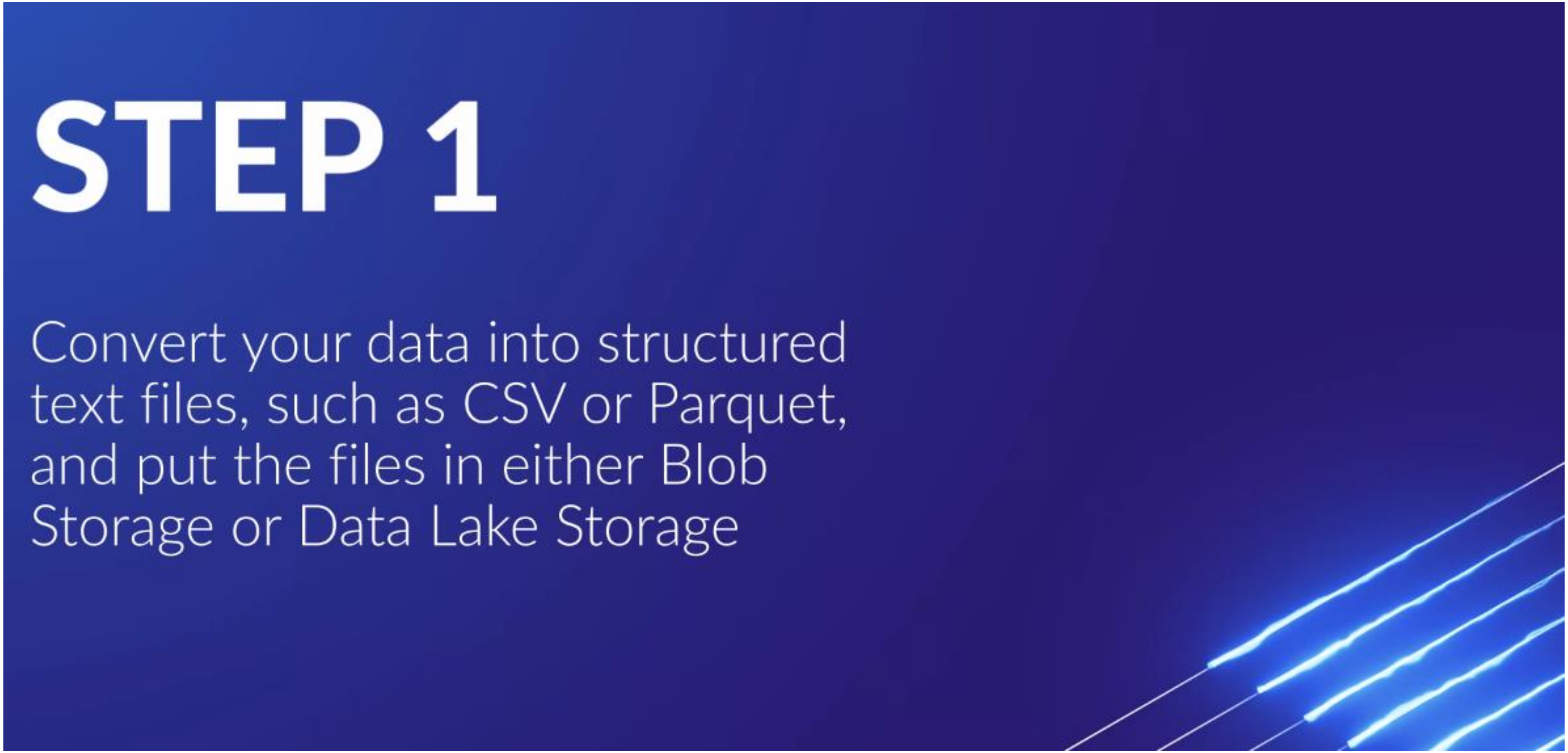
If a relatively small dimension table is frequently used in joins and aggregations, then it will be much more efficient to have it on every distribution



Synapse Analytics

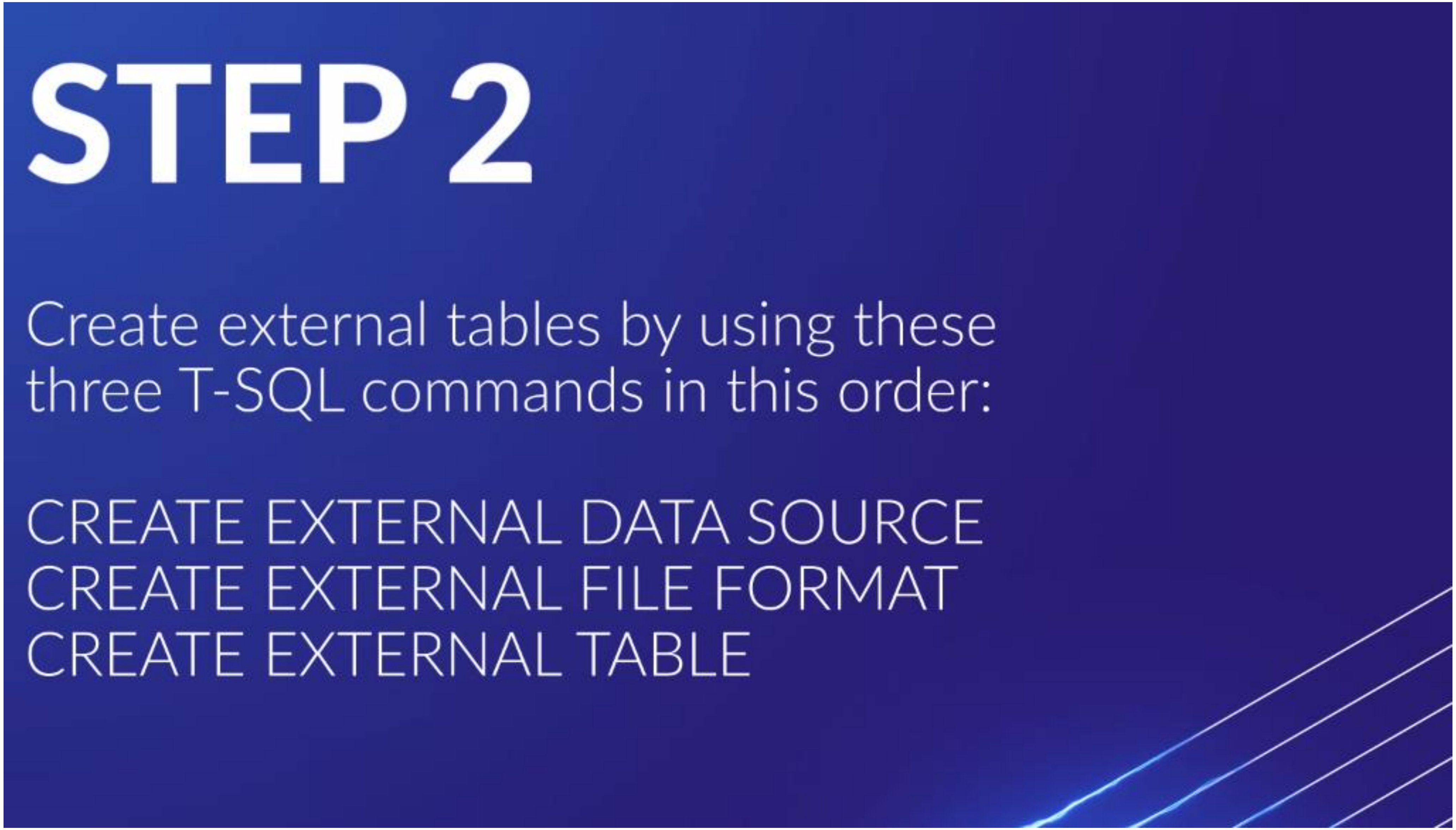
The fastest way to load data into **Synapse Analytics** is to use **Polybase**

It lets you read data from external sources using **T-SQL**



STEP 1

Convert your data into structured text files, such as CSV or Parquet, and put the files in either Blob Storage or Data Lake Storage



STEP 2

Create external tables by using these three T-SQL commands in this order:

```
CREATE EXTERNAL DATA SOURCE
CREATE EXTERNAL FILE FORMAT
CREATE EXTERNAL TABLE
```

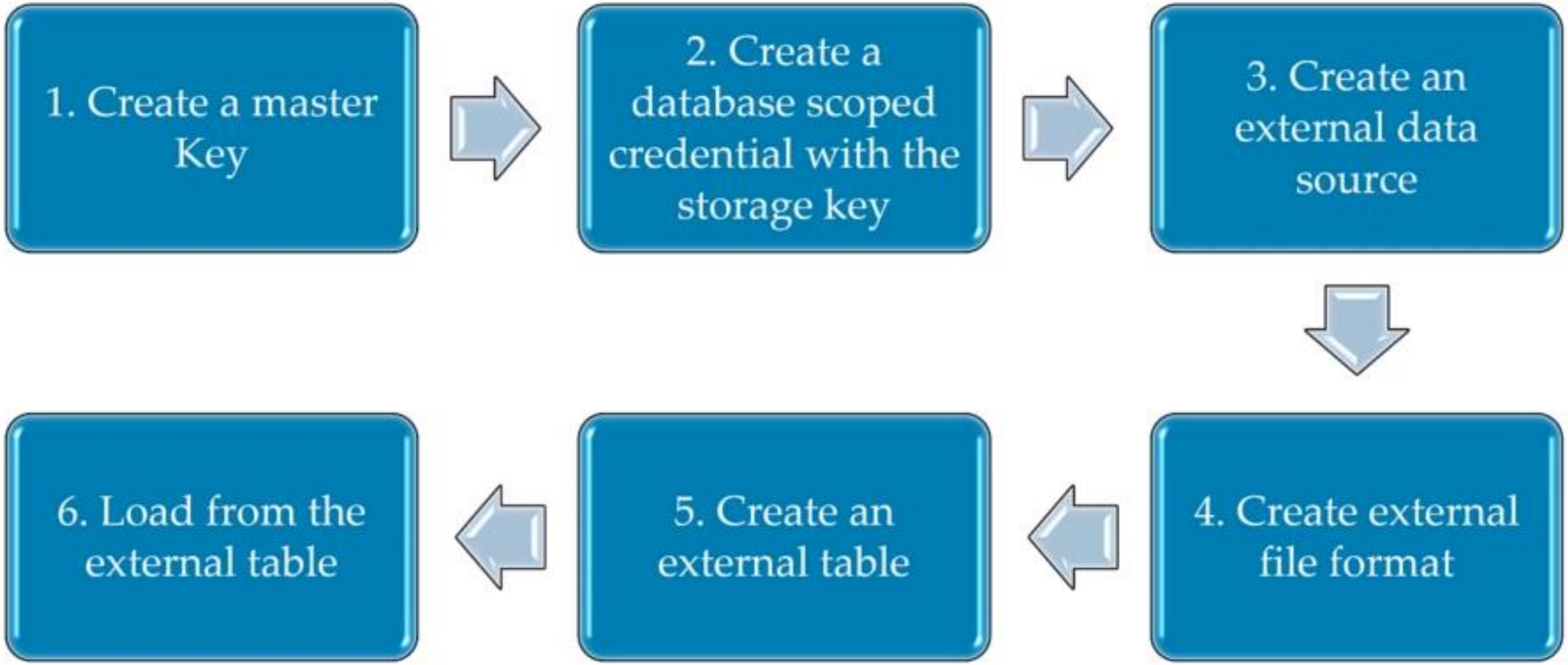
STEP 3

Load the data into a staging table
in Synapse Analytics

STEP 4

Insert the data into production tables

PolyBase Setup



1. Create a master Key

2. Create a database scoped credential with the storage key

3. Create an external data source

4. Create external file format

5. Create an external table

6. Load from the external table

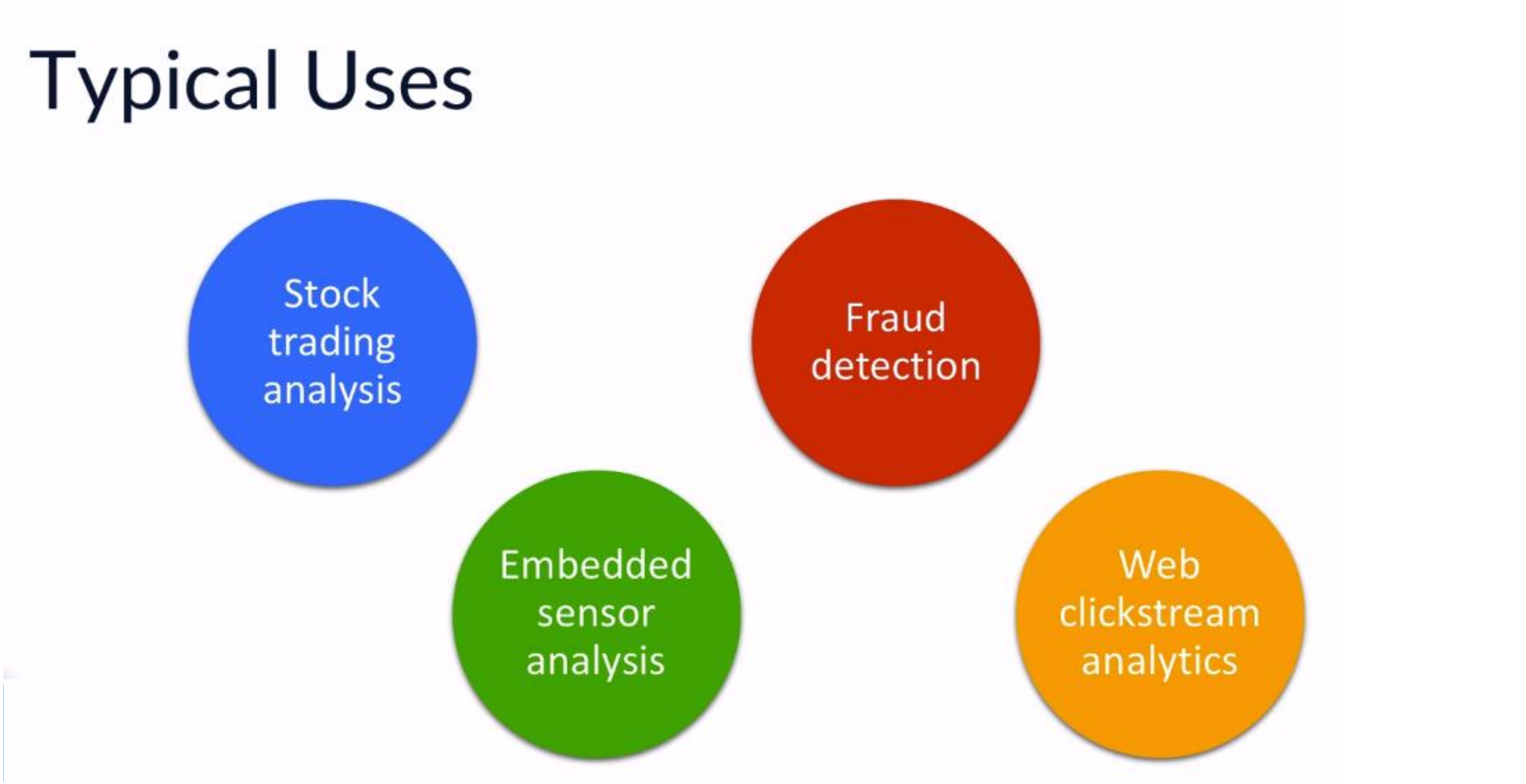
Stream Analytics

The Rise of Streaming Analytics

Batch processing stopped working because:

Batch jobs were taking longer and needed more processing power

Companies had applications that needed data to be processed in real time



Typical Uses

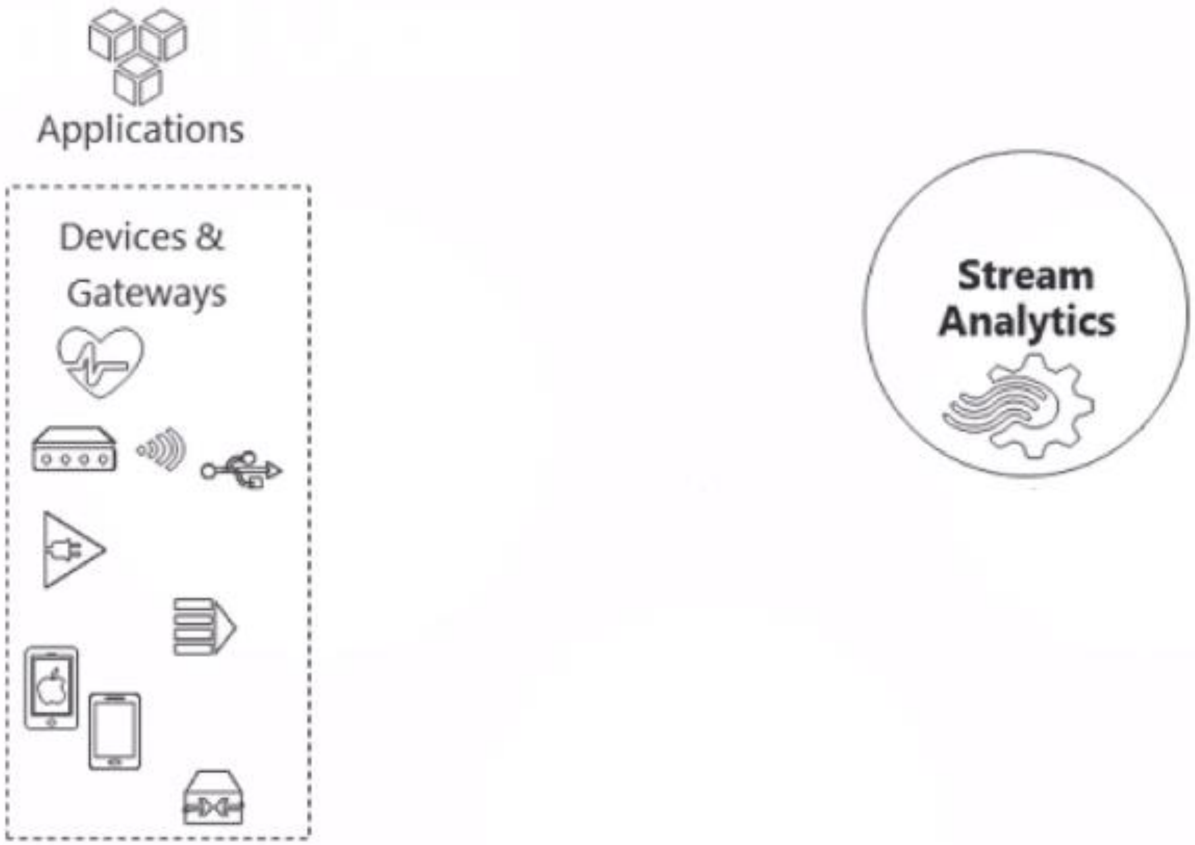
Stock trading analysis

Fraud detection

Embedded sensor analysis

Web clickstream analytics

Stream
Analytics

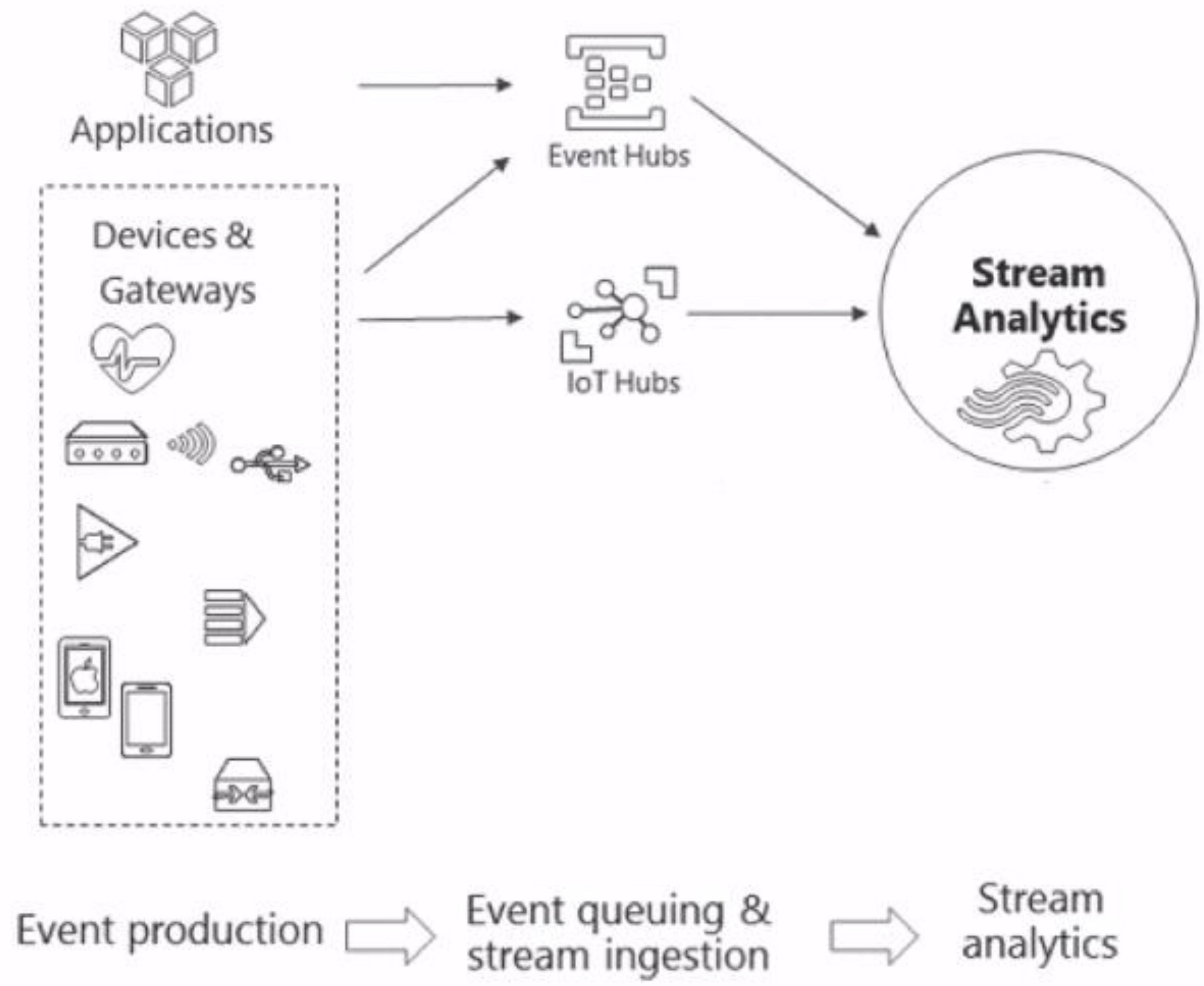


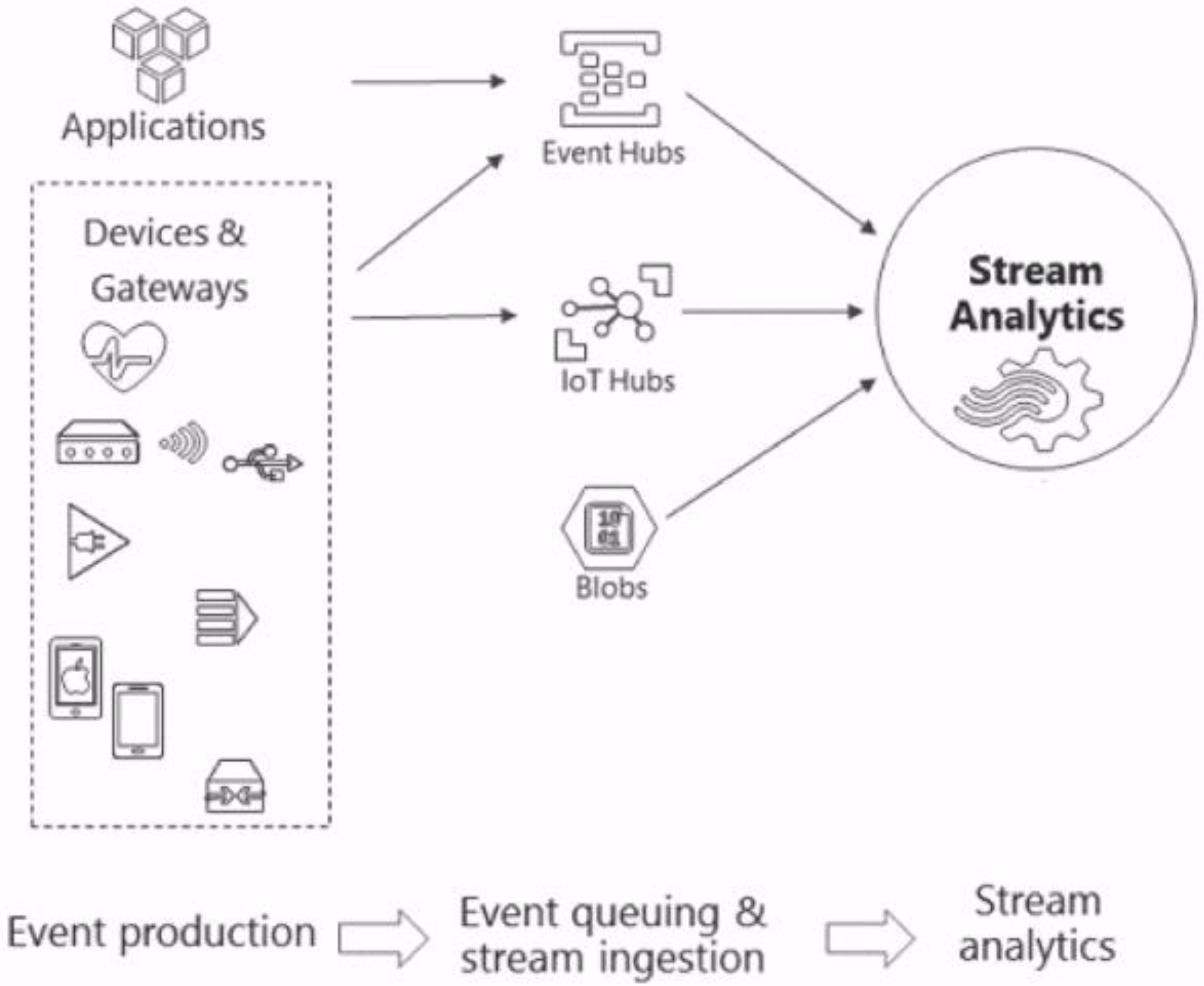
Applications

Devices &
Gateways

Stream
Analytics

Event production


Applications
Devices &
Gateways
Event Hubs
IoT Hubs
Stream
Analytics
Event production
Event queuing &
stream ingestion
Stream
analytics



Applications

Devices & Gateways

Event Hubs

IoT Hubs

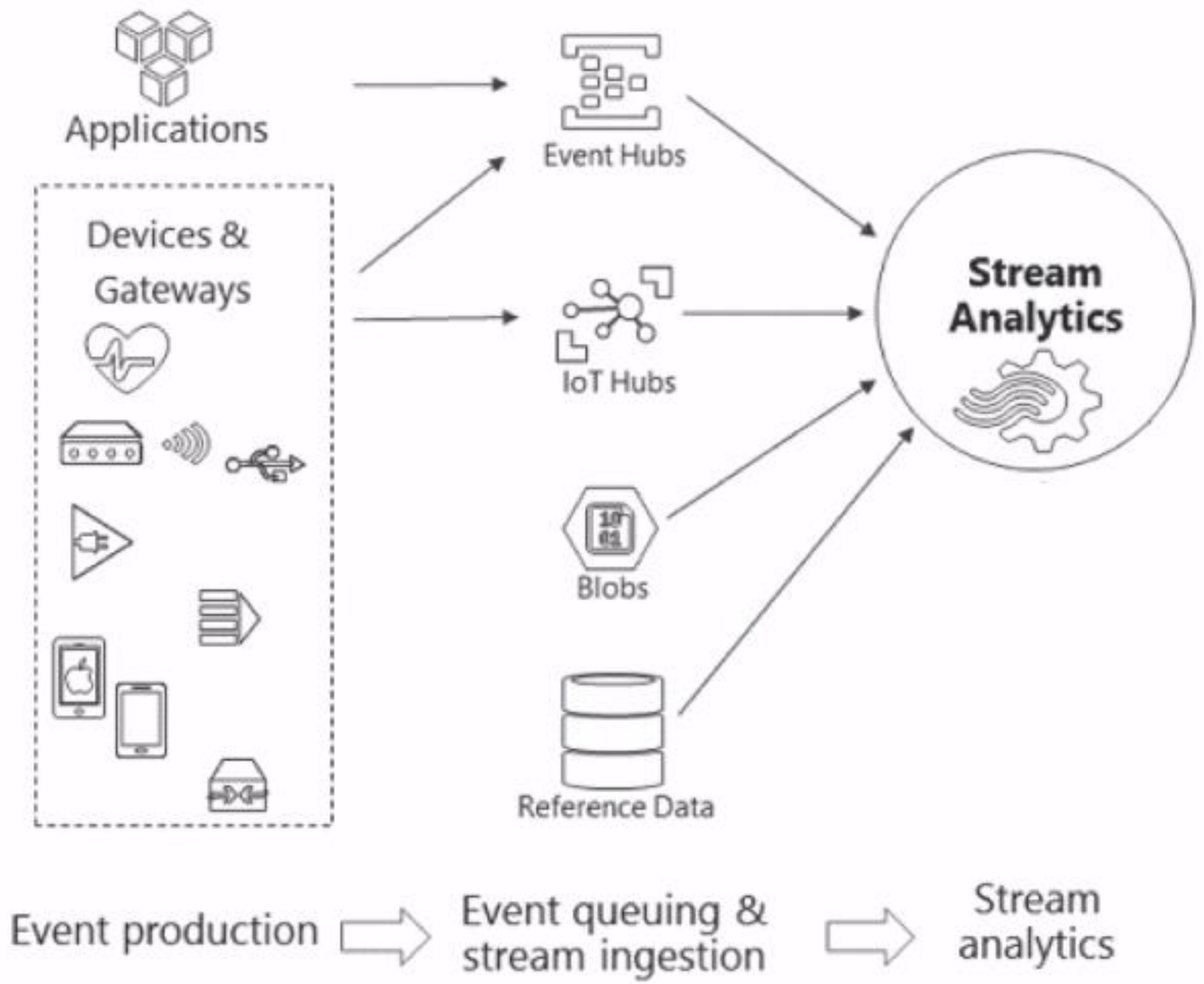
Blobs

Stream Analytics

Event production

Event queuing & stream ingestion

Stream analytics



Applications

Devices & Gateways

Event Hubs

IoT Hubs

Blobs

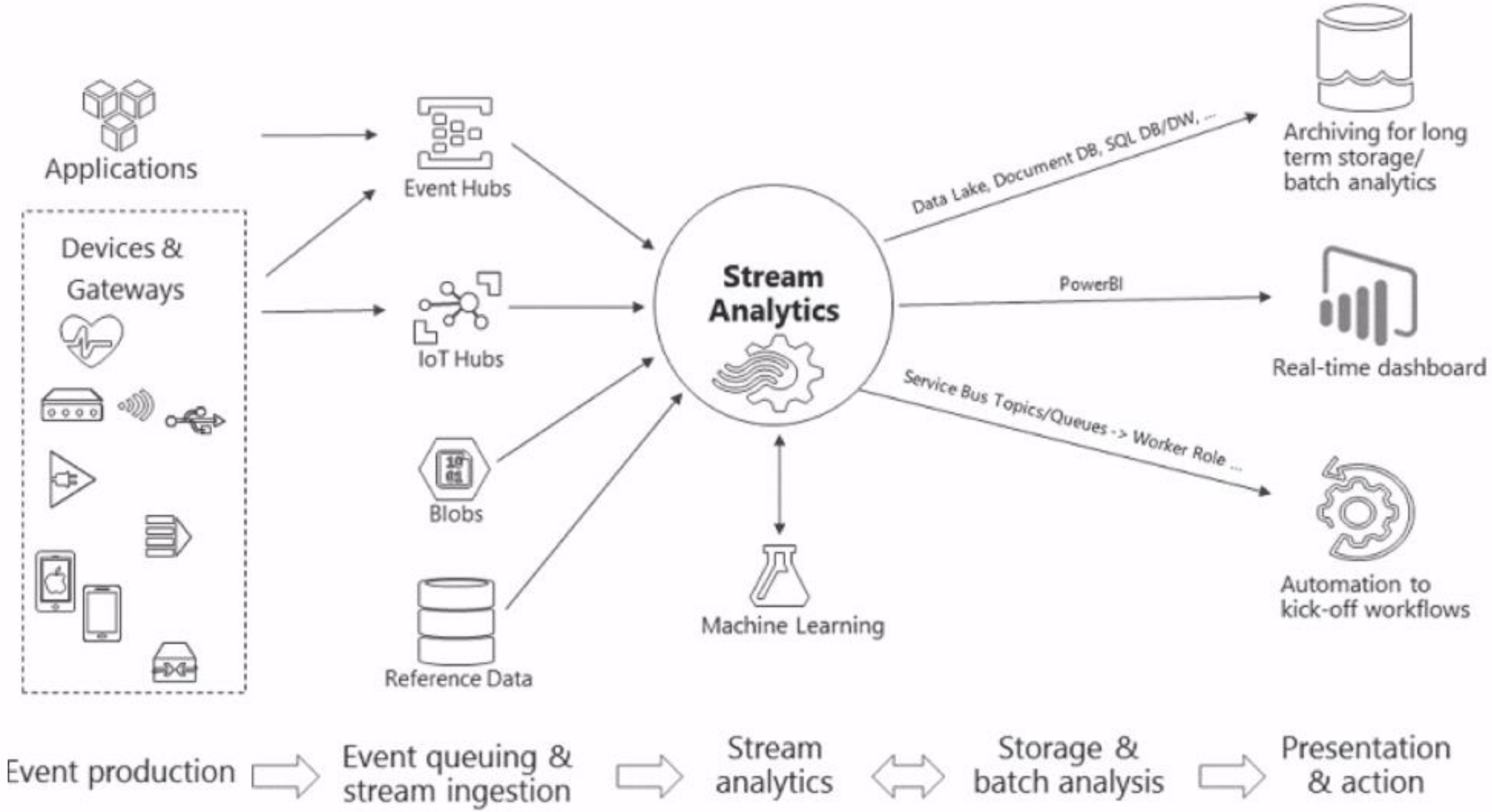
Reference Data

Stream Analytics

Event production

Event queuing & stream ingestion

Stream analytics



Applications

Devices & Gateways

Event Hubs

IoT Hubs

Blobs

Reference Data

Stream Analytics

Machine Learning

Data Lake, Document DB, SQL DB/DW, ...

PowerBI

Service Bus Topics/Queues -> Worker Role ...

Archiving for long term storage/ batch analytics

Real-time dashboard

Automation to kick-off workflows

Event production ⇨ Event queuing & stream ingestion ⇨ Stream analytics ⇔ Storage & batch analysis ⇨ Presentation & action

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

Alternatives

Spark or Storm on HDInsight

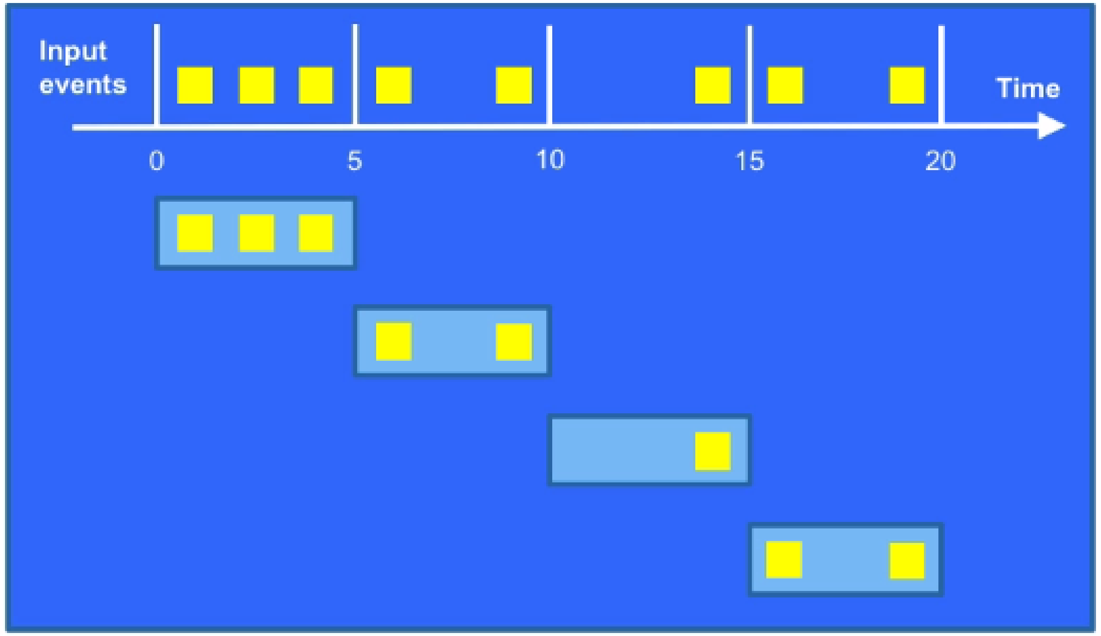
- Cluster
- Hadoop ecosystem
- Scala, Java, or Python

Azure Streaming Analytics

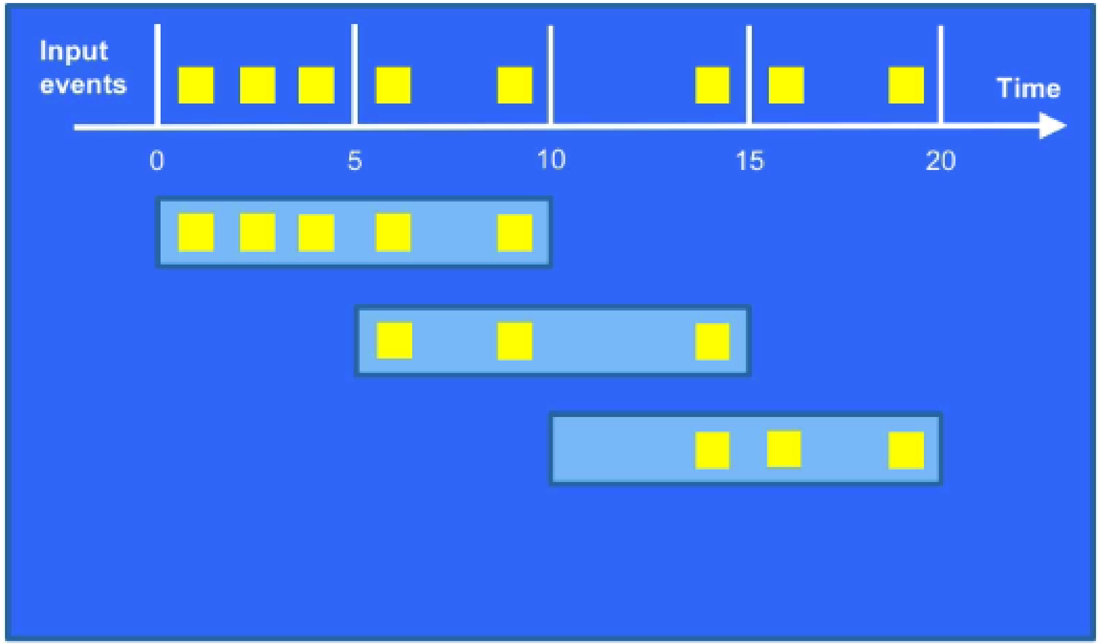
- Job
- Microsoft ecosystem
- Stream Analytics Query Language (variant of T-SQL)

Time Windows

Temporal Windows

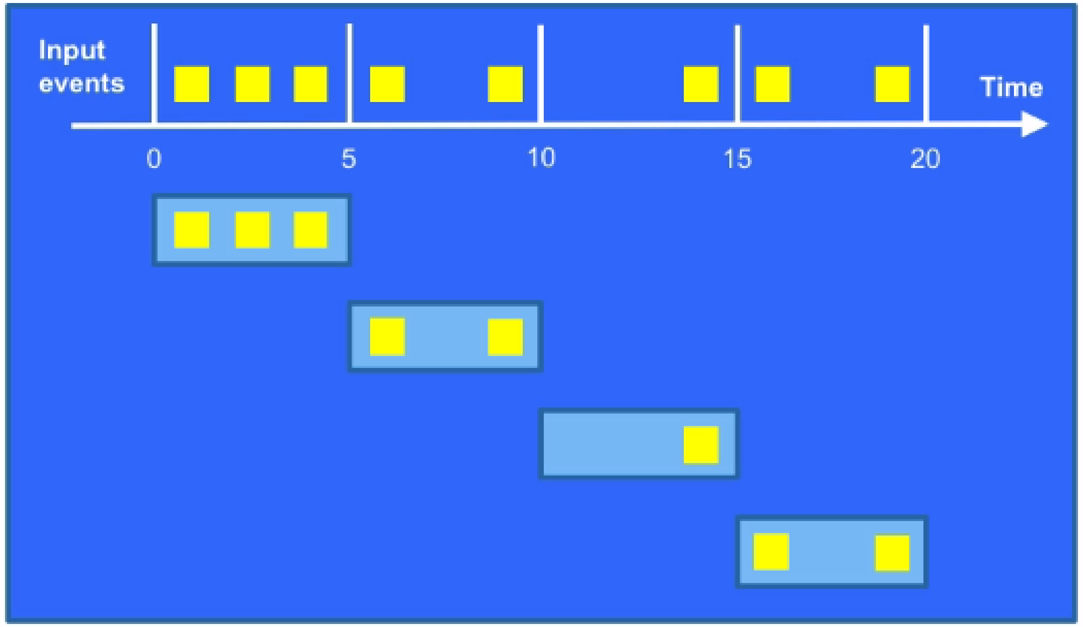


Tumbling Windows

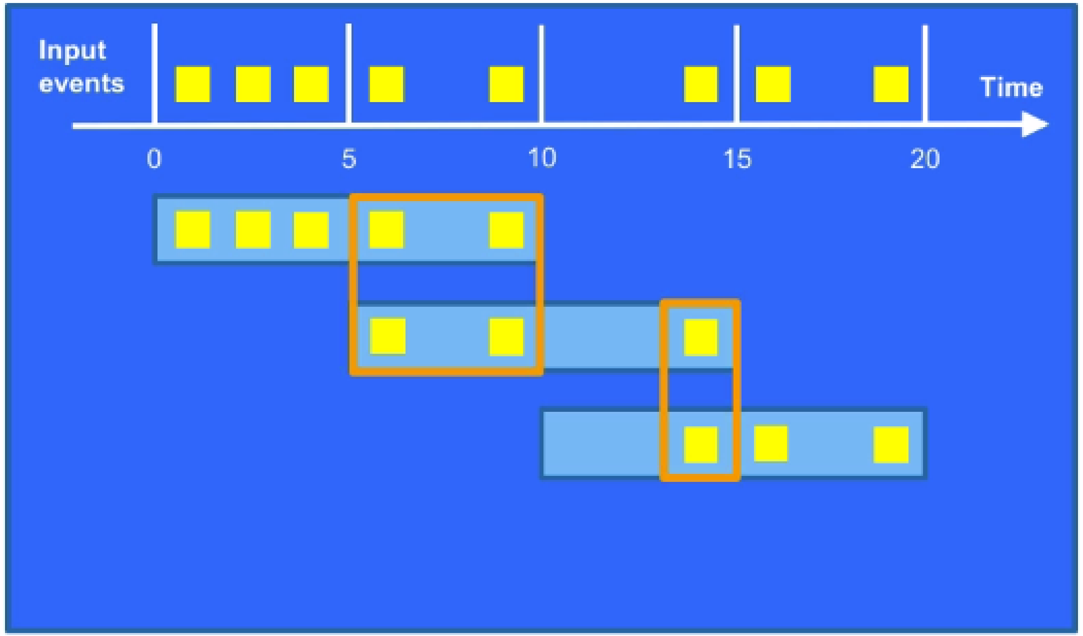


Hopping Windows

Temporal Windows

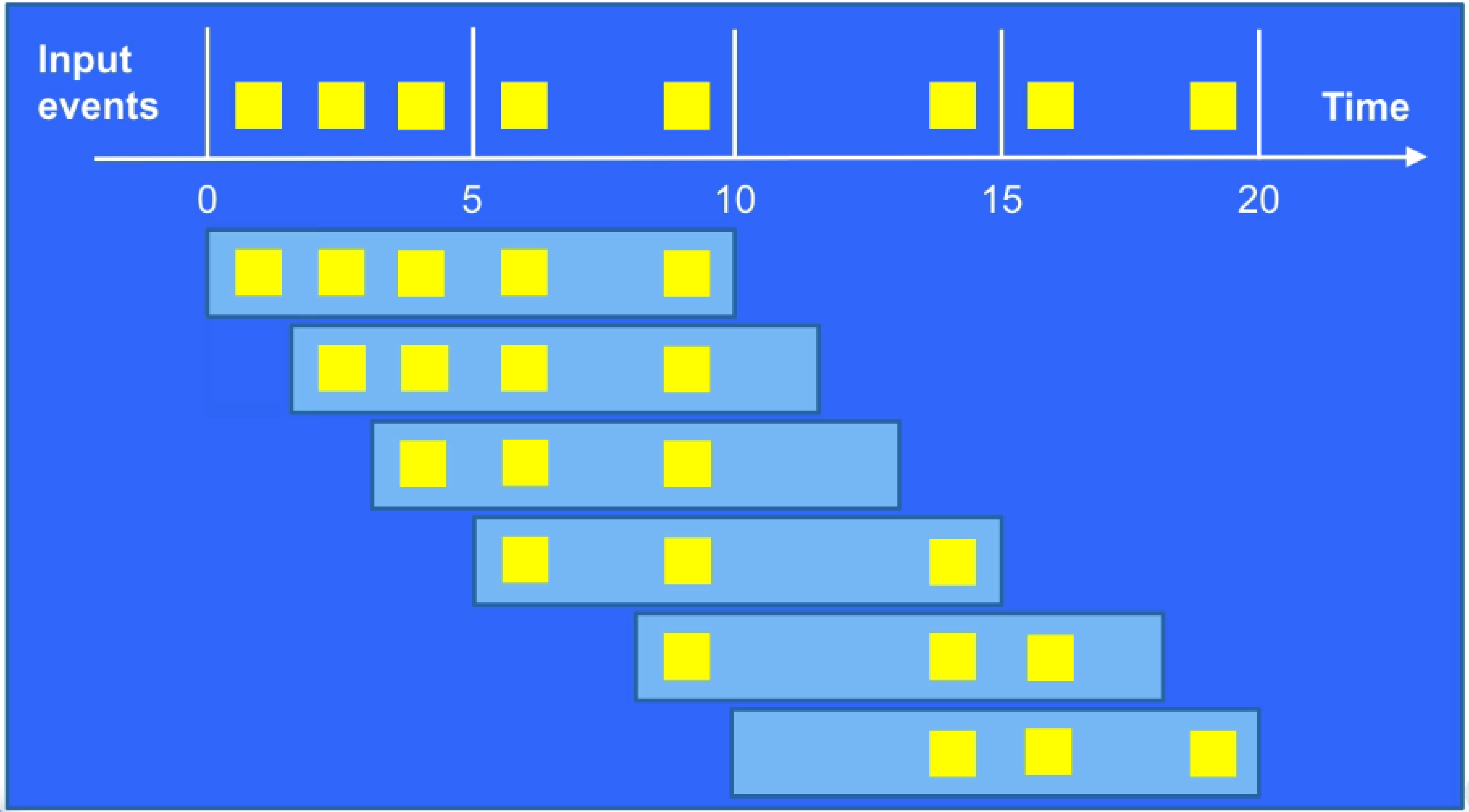


Tumbling Windows

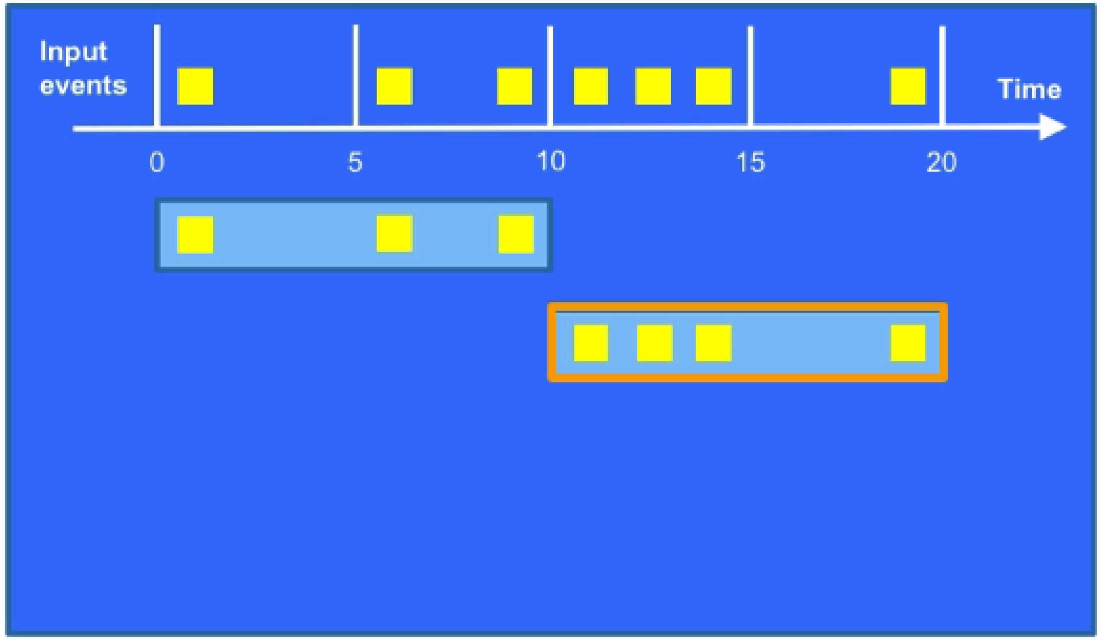


Hopping Windows

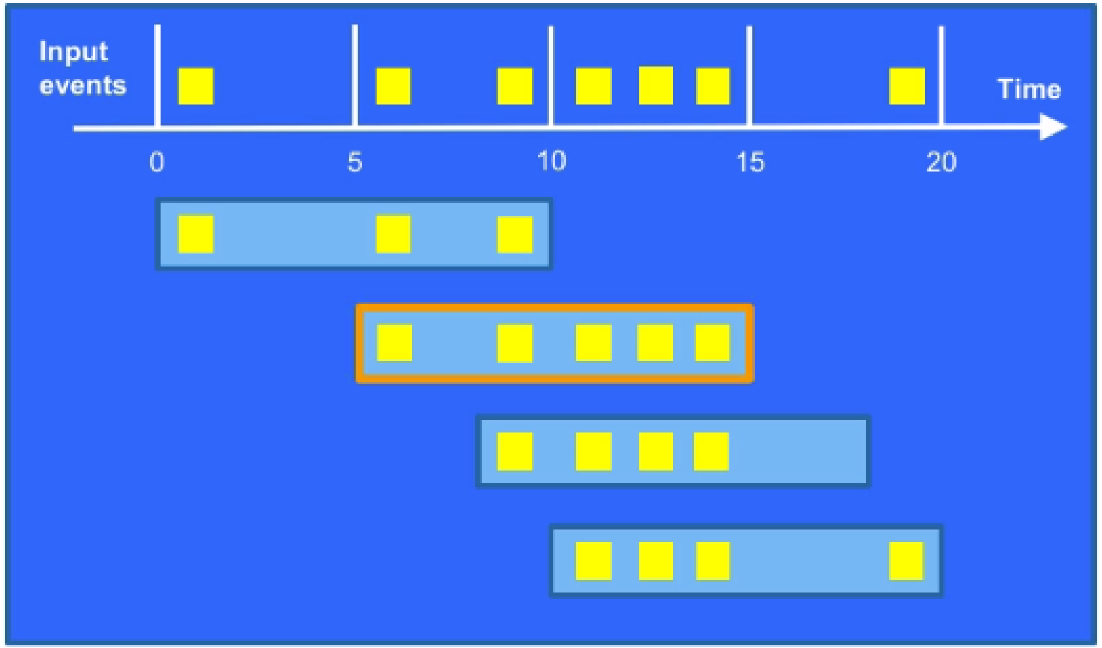
Sliding Windows



Tumbling vs. Sliding Windows

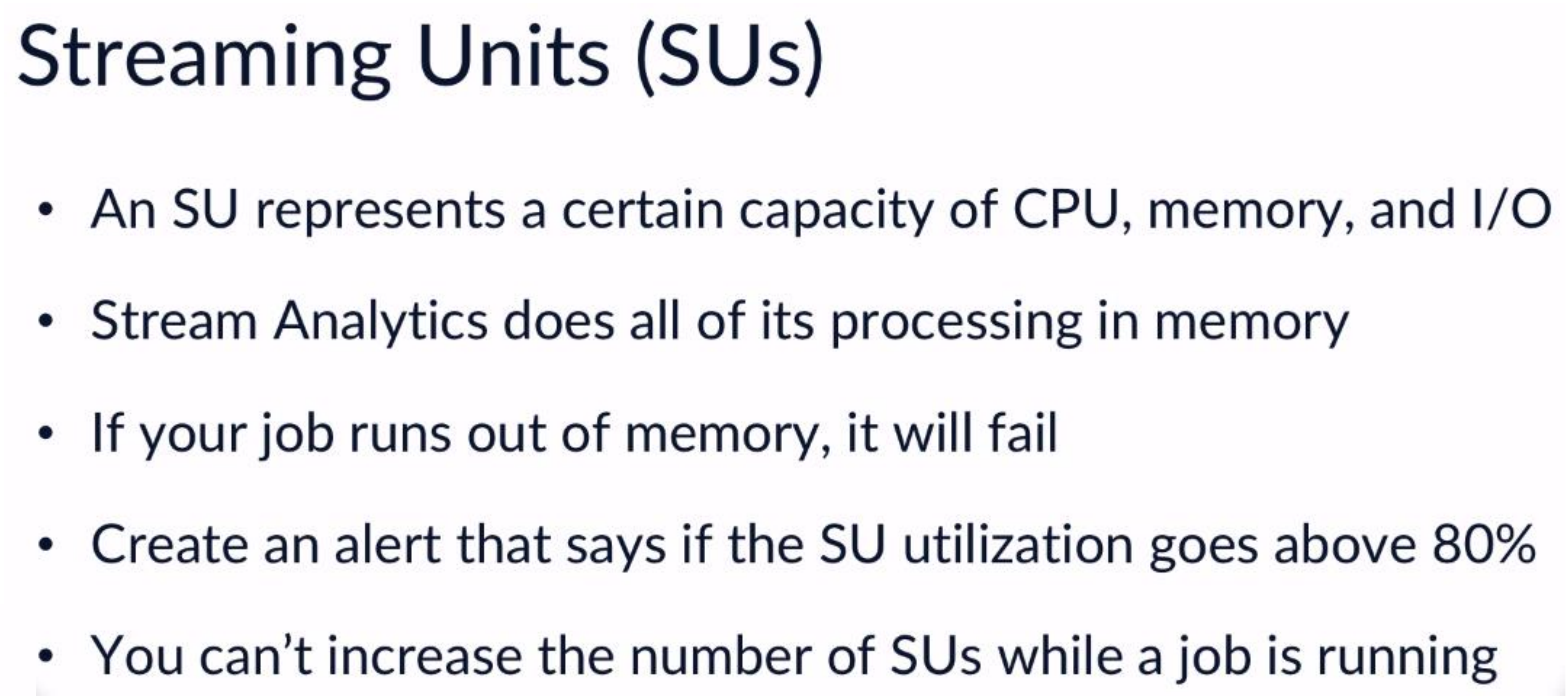


Tumbling Windows



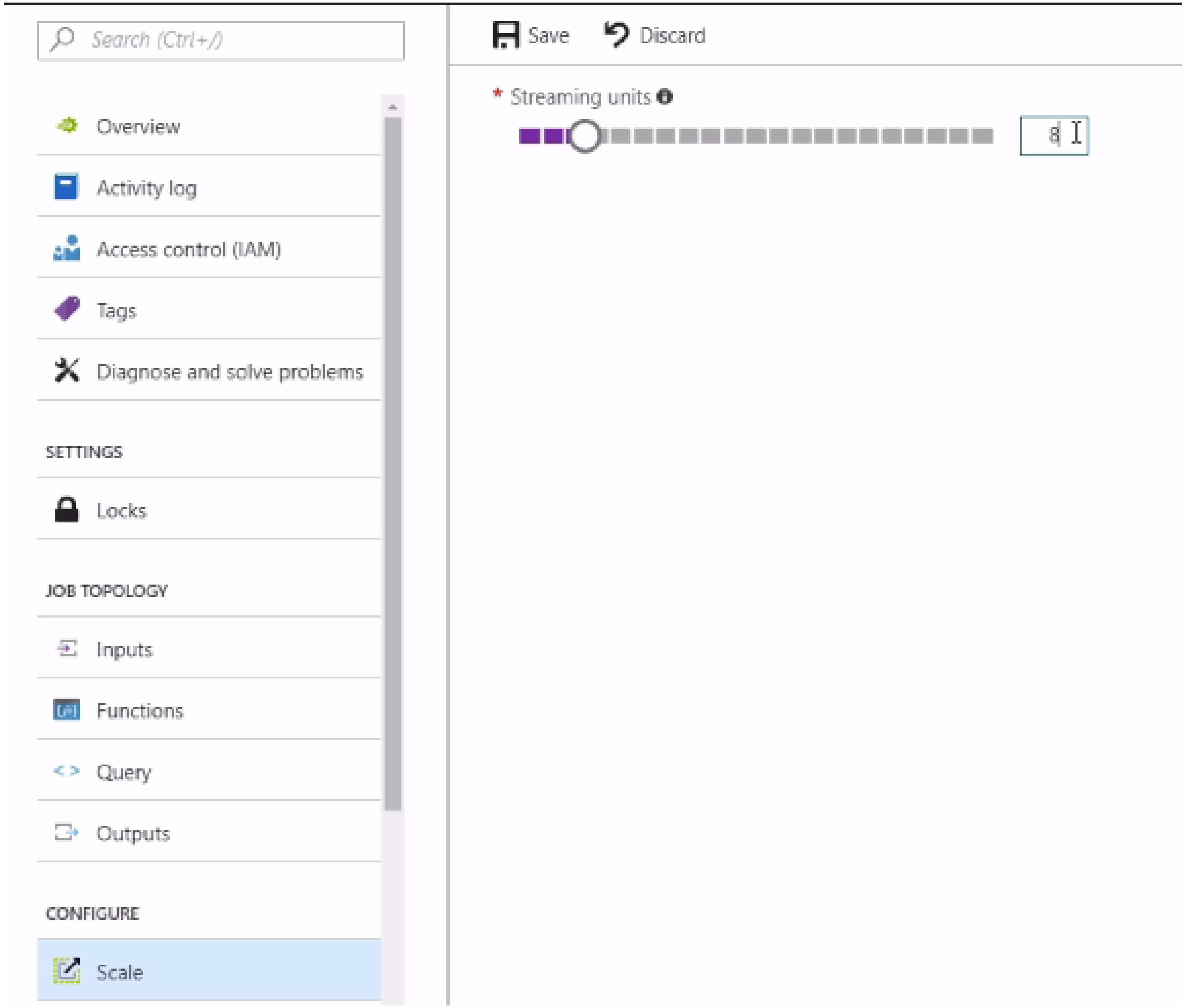
Sliding Windows

Scaling



Streaming Units (SUs)

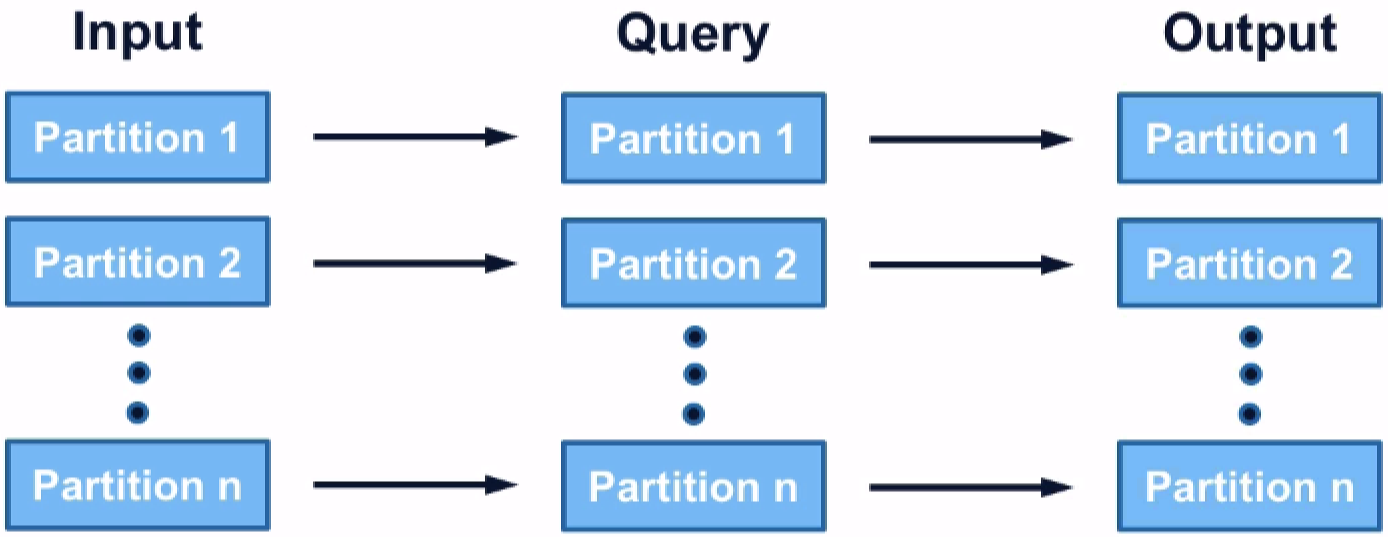
- An SU represents a certain capacity of CPU, memory, and I/O
- Stream Analytics does all of its processing in memory
- If your job runs out of memory, it will fail
- Create an alert that says if the SU utilization goes above 80%
- You can’t increase the number of SUs while a job is running

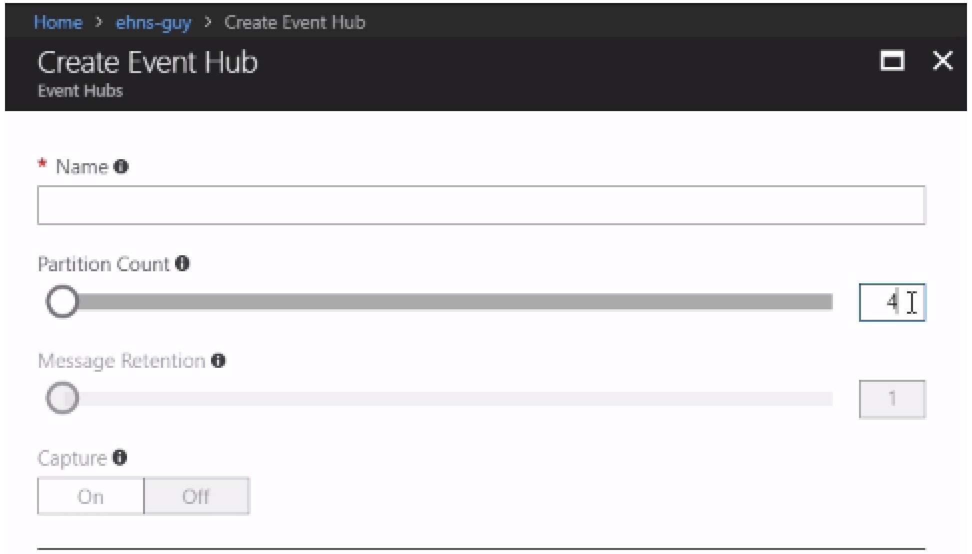


* Streaming units 8

Parallelizing a Job

- Embarrassingly parallel
  - The whole job can be split into parallel tasks for multiple workers





Create Event Hub

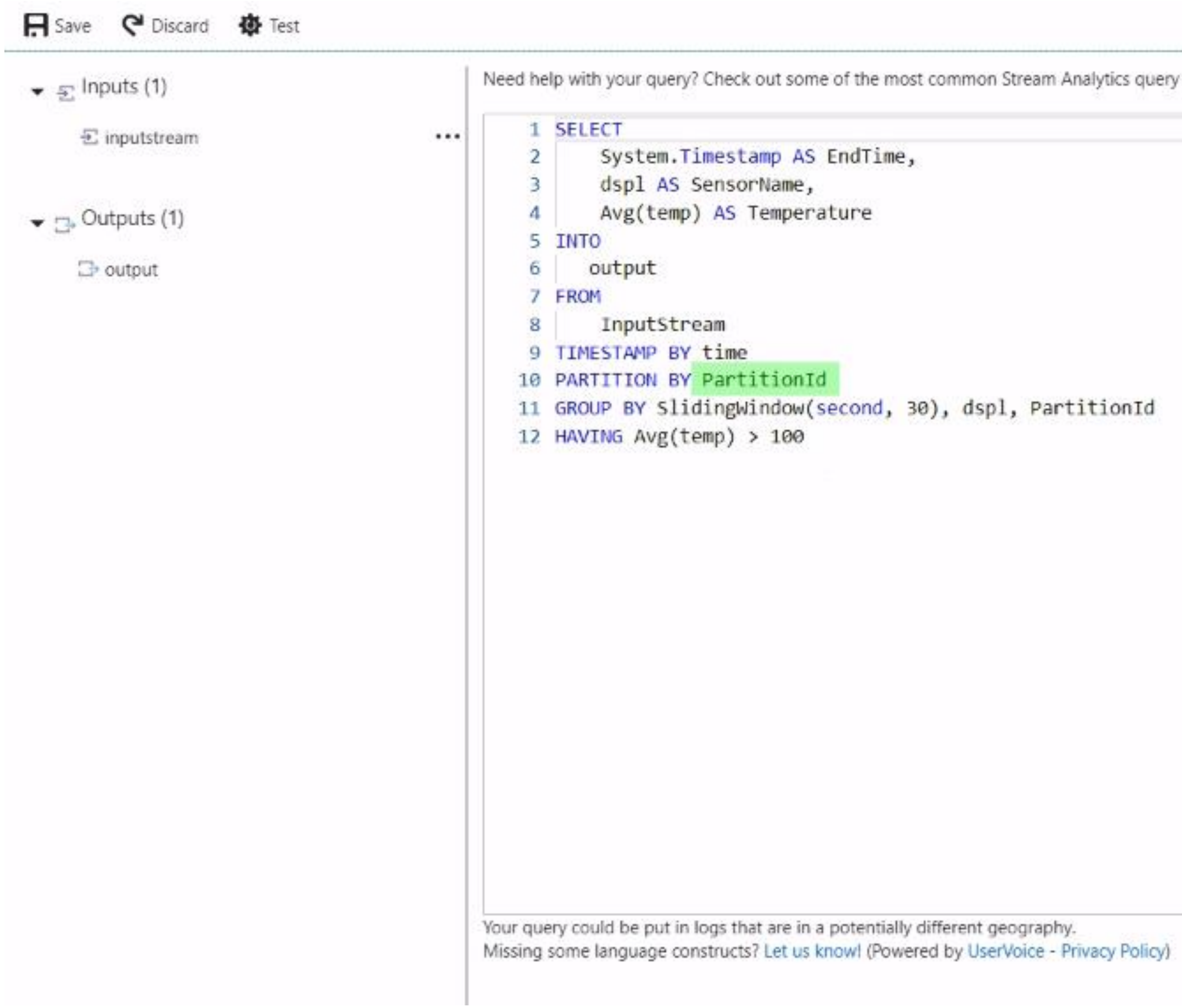
Event Hubs

* Name

Partition Count: 4

Message Retention: 1

Capture: On Off



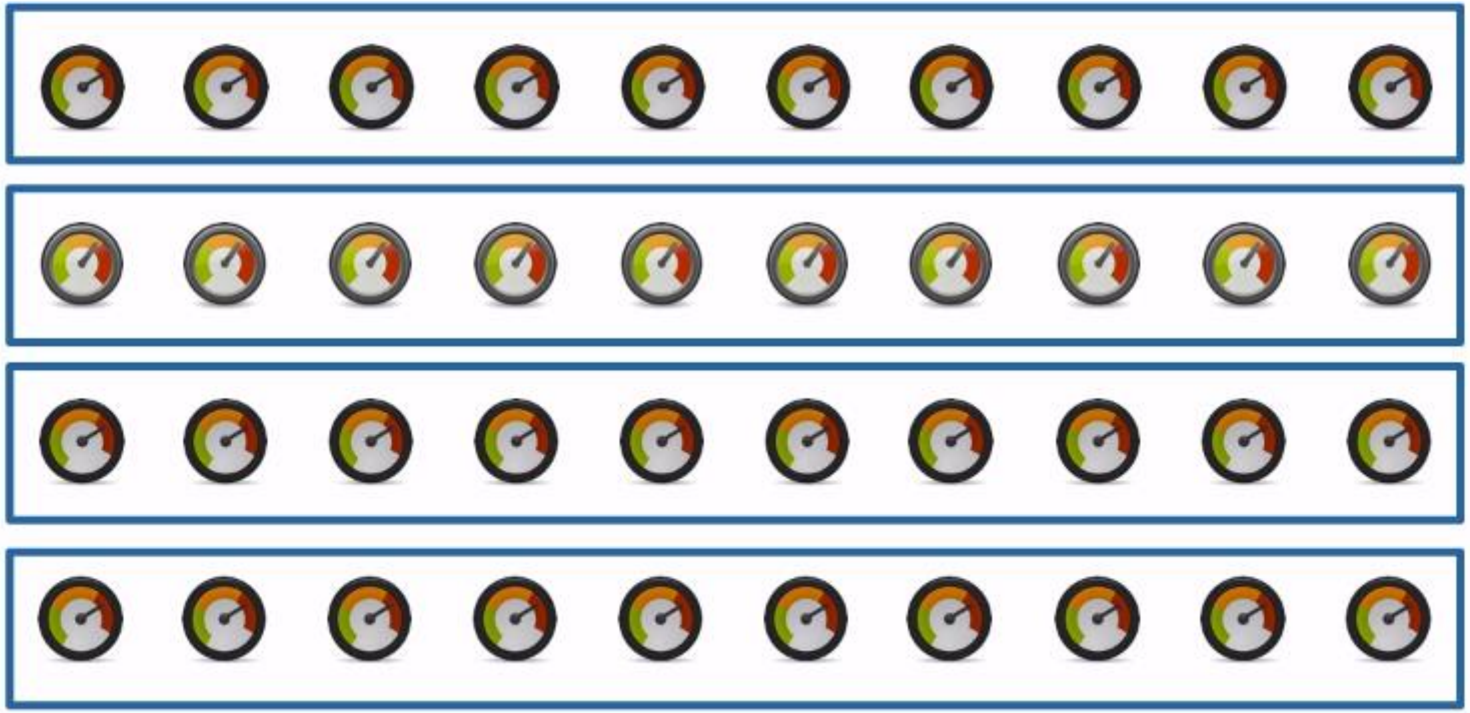
Need help with your query? Check out some of the most common Stream Analytics query

```
SELECT
    System.Timestamp AS EndTime,
    dspl AS SensorName,
    Avg(temp) AS Temperature
INTO
   output
FROM
    InputStream
TIMESTAMP BY time
PARTITION BY PartitionId
GROUP BY SlidingWindow(second, 30), dspl, PartitionId
HAVING Avg(temp) > 100
```

Your query could be put in logs that are in a potentially different geography.
Missing some language constructs? Let us know! (Powered by UserVoice - Privacy Policy)

Partitioning Input

Partitioning Input



Output Destinations

Power BI, SQL Database, SQL Data Warehouse

- Don't support partitioning

Event Hub, IoT Hub, Cosmos DB

- Need to set the partition key

Data Lake Store, Functions, Table or Blob Storage, Service Bus

How Many SUs to Allocate

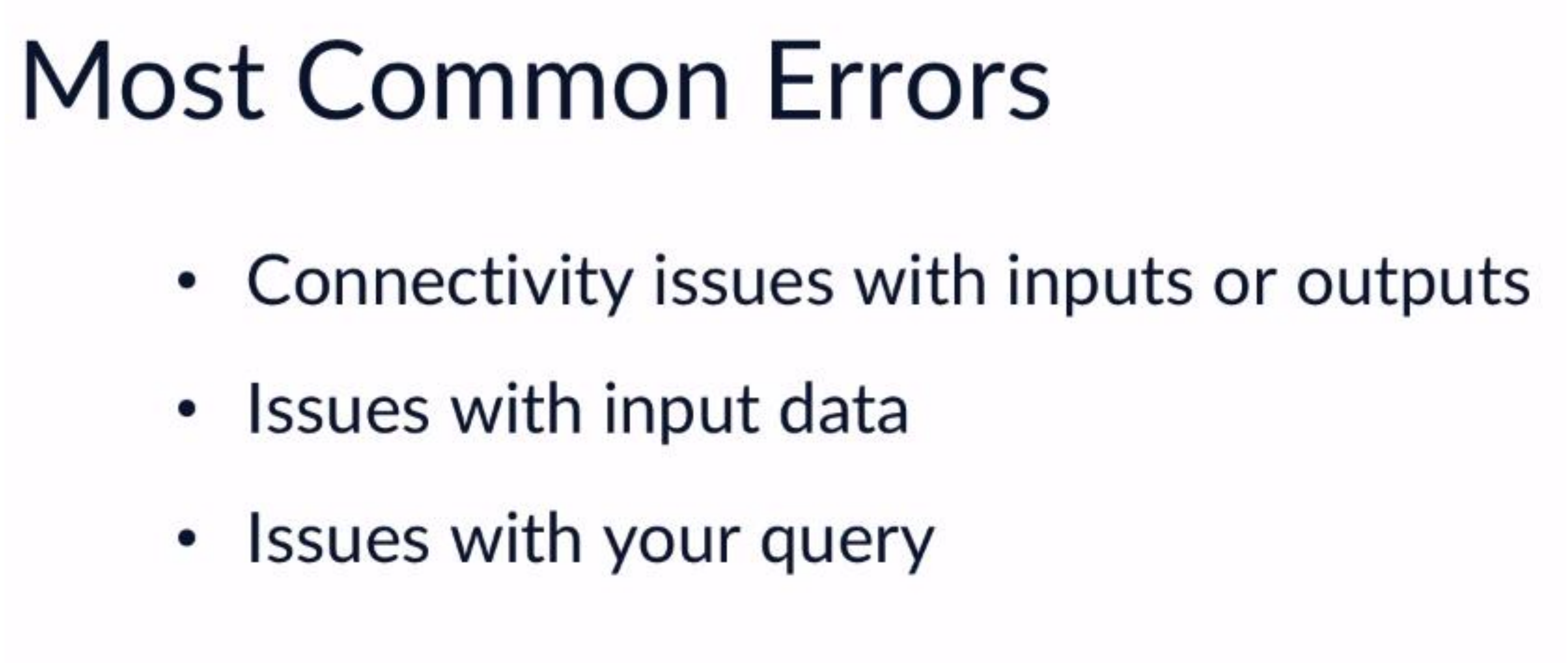
- 6 SU represents the full capacity of a single computing node
- If job isn’t parallelizable, 6 SU gives maximum performance
- If you break query into multiple steps, Stream Analytics will put each step on its own 6 SU node
- If job isn’t parallelizable, try running it with

  6 SU x number of steps in query

How Many SUs to Allocate

- If the job is parallelizable and it’s too big to run on 6 SUs
  - Partition the input and output
  - Use PARTITION BY in your query
  - Try 6 SU x number of partitions
- # of partitions should be evenly divisible by # of nodes
  - For 4 partitions, choose either 2 or 4 nodes, but not 3

Troubleshooting

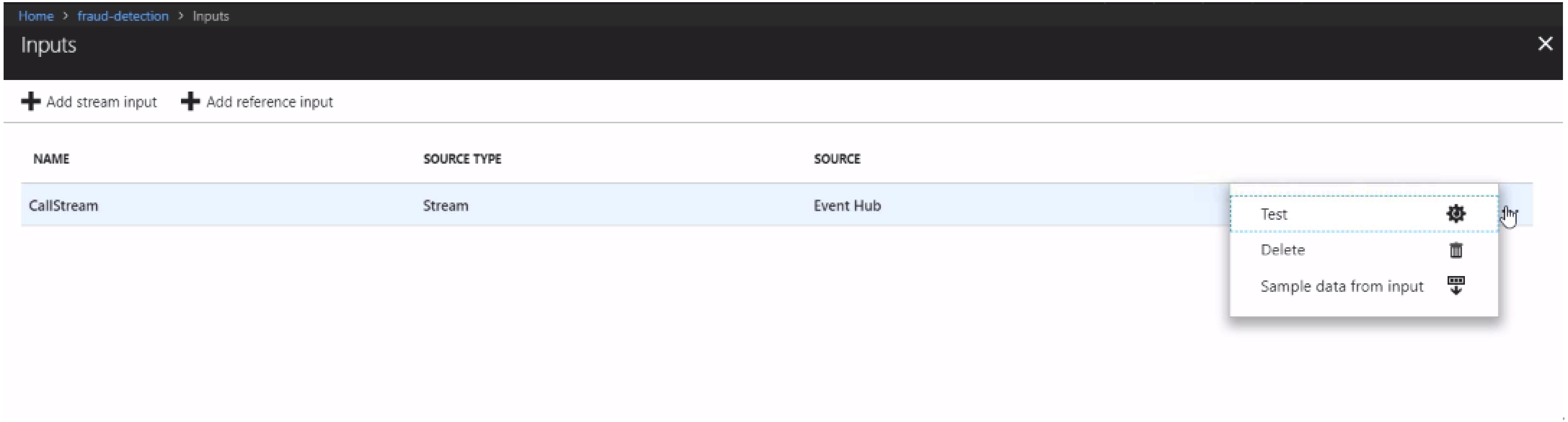


Most Common Errors

- Connectivity issues with inputs or outputs
- Issues with input data
- Issues with your query

Most Common Errors

- Connectivity issues with inputs or outputs
  - Use the Test feature



Home > fraud-detection > Inputs

Inputs

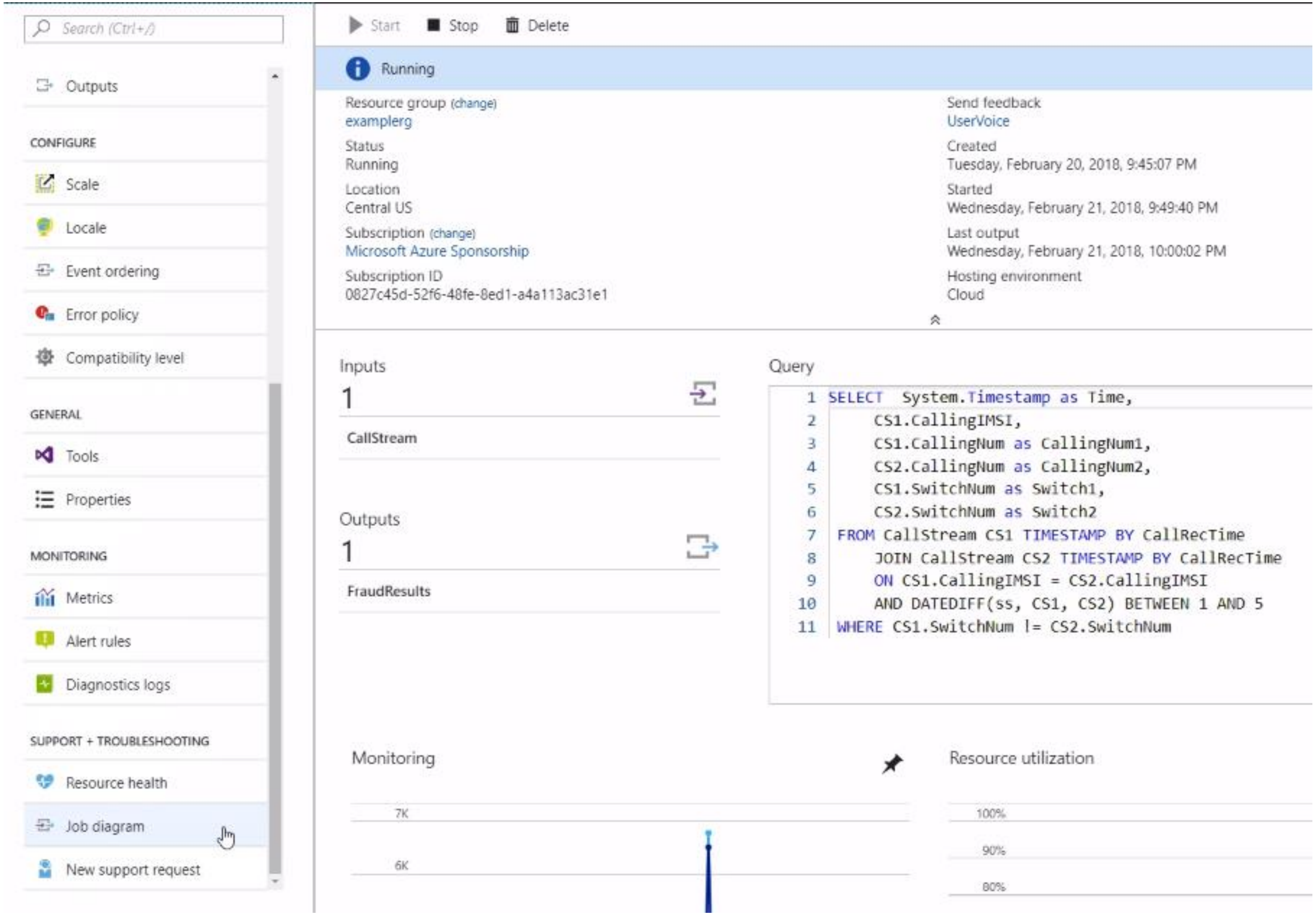
NAME	SOURCE TYPE	SOURCE
CallStream	Stream	Event Hub

Most Common Errors

- Issues with input data
  - Stop the job
  - Change the query to “SELECT * FROM InputStream
  - Then run a test by sampling the data

Most Common Errors

- Issues with your query
  - Reduce it to a simpler query, test it, and then build it back up, testing at every step
  - Common problem #1: Having a WHERE clause that filters out every input record, so there are no outputs
  - Common problem #2: Timestamps are earlier than the job start time, so all of the input records are dropped



Resource group (change)
examplerg

Status
Running

Location
Central US

Subscription (change)
Microsoft Azure Sponsorship

Subscription ID
0827c45d-52f6-48fe-8ed1-a4a113ac31e1

Send feedback
UserVoice

Created
Tuesday, February 20, 2018, 9:45:07 PM

Started
Wednesday, February 21, 2018, 9:49:40 PM

Last output
Wednesday, February 21, 2018, 10:00:02 PM

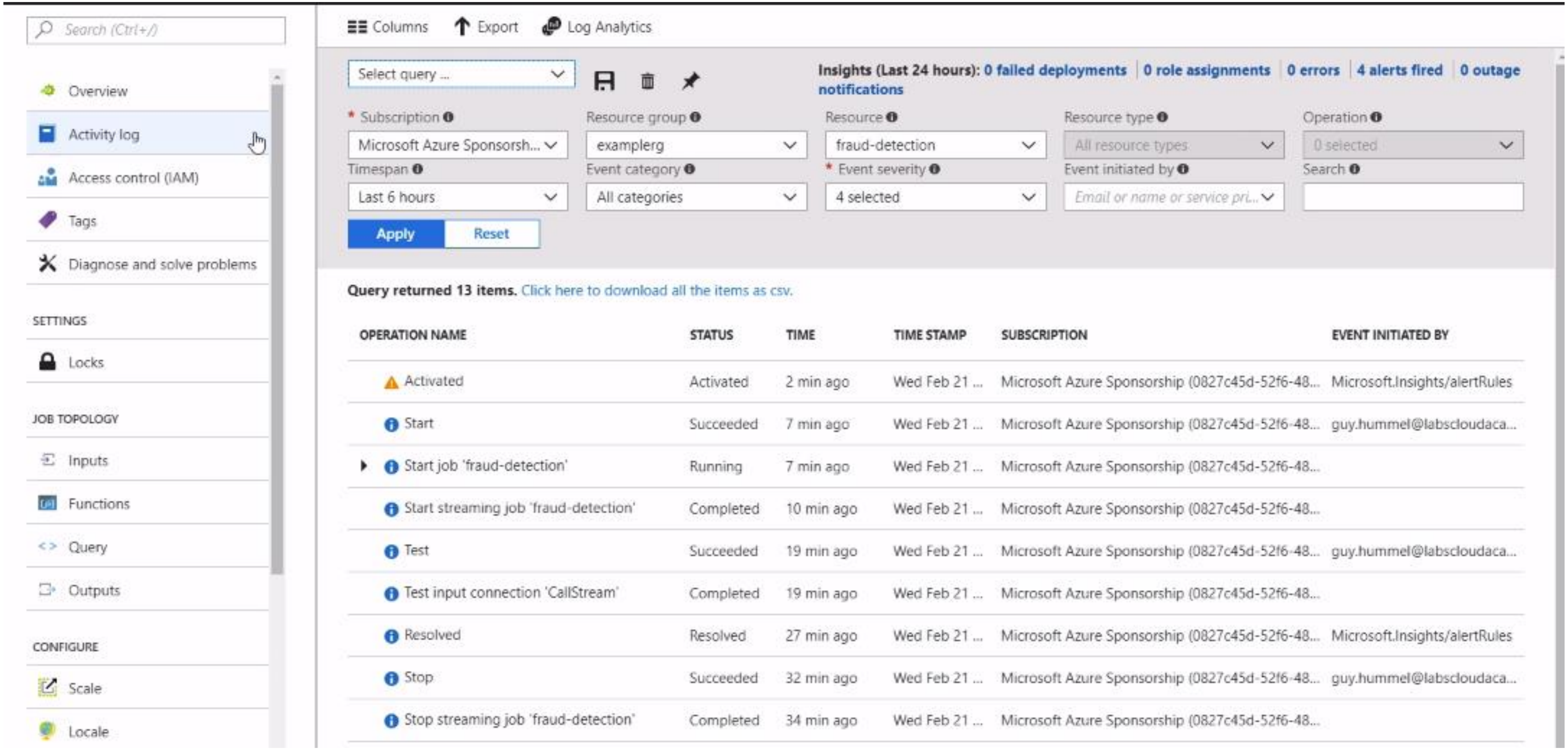
Hosting environment
Cloud

Inputs
1
CallStream

Outputs
1
FraudResults

Query

```
SELECT  System.Timestamp as Time,
    CS1.CallingIMSI,
    CS1.CallingNum as CallingNum1,
    CS2.CallingNum as CallingNum2,
    CS1.SwitchNum as Switch1,
    CS2.SwitchNum as Switch2
 FROM CallStream CS1 TIMESTAMP BY CallRecTime
    JOIN CallStream CS2 TIMESTAMP BY CallRecTime
    ON CS1.CallingIMSI = CS2.CallingIMSI
    AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5
 WHERE CS1.SwitchNum != CS2.SwitchNum
```

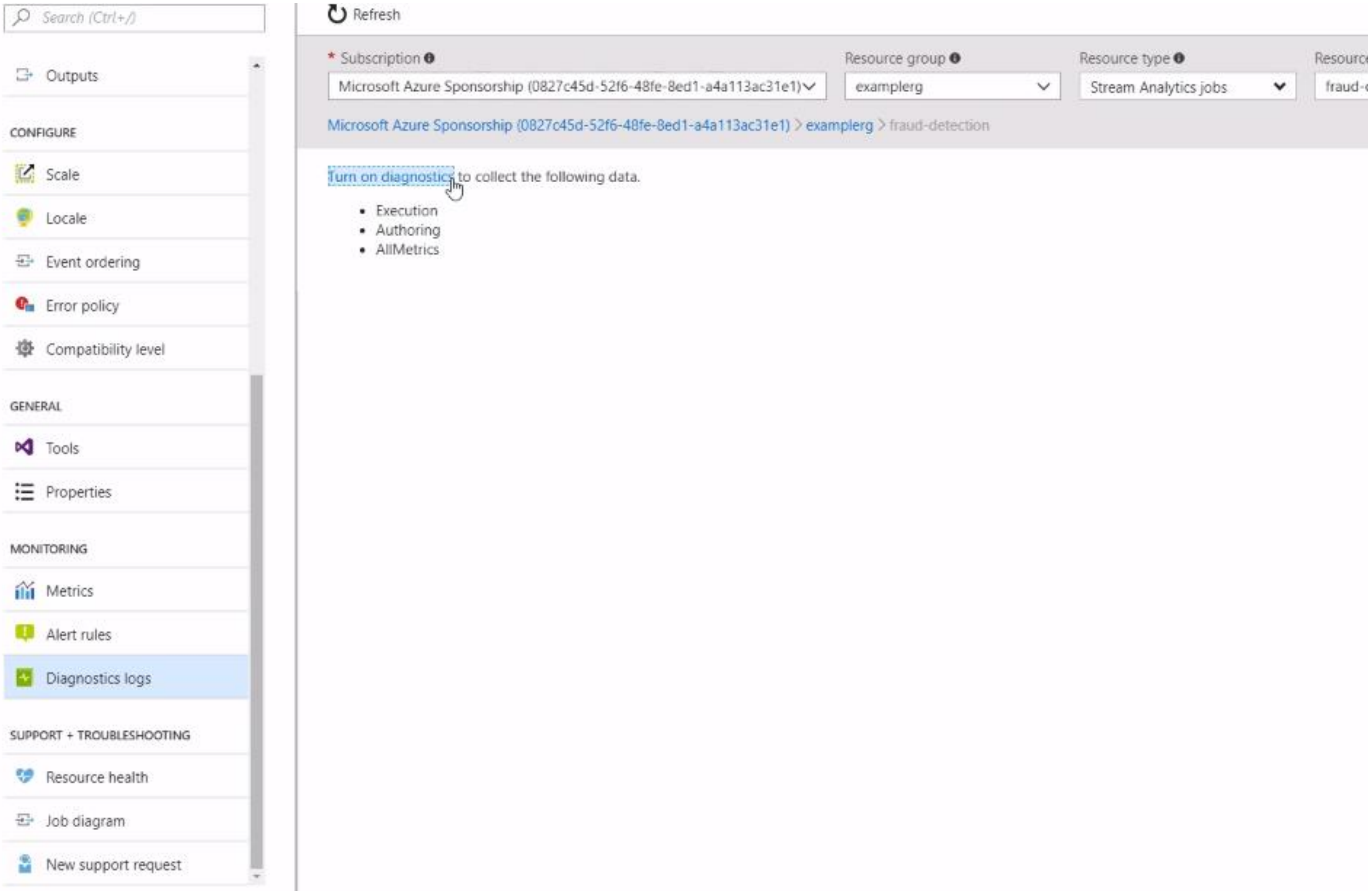


Insights (Last 24 hours): 0 failed deployments | 0 role assignments | 0 errors | 4 alerts fired | 0 outage notifications

* Subscription: Microsoft Azure Sponsorsh...
Resource group: examplerg
Resource: fraud-detection
Resource type: All resource types
Operation: 0 selected
Timespan: Last 6 hours
Event category: All categories
* Event severity: 4 selected
Event initiated by: Email or name or service pri...
Search:

Query returned 13 items. Click here to download all the items as csv.

OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION	EVENT INITIATED BY
⚠ Activated	Activated	2 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	Microsoft.Insights/alertRules
ⓘ Start	Succeeded	7 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	guy.hummel@labscloudaca...
▸ ⓘ Start job 'fraud-detection'	Running	7 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	
ⓘ Start streaming job 'fraud-detection'	Completed	10 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	
ⓘ Test	Succeeded	19 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	guy.hummel@labscloudaca...
ⓘ Test input connection 'CallStream'	Completed	19 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	
ⓘ Resolved	Resolved	27 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	Microsoft.Insights/alertRules
ⓘ Stop	Succeeded	32 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	guy.hummel@labscloudaca...
ⓘ Stop streaming job 'fraud-detection'	Completed	34 min ago	Wed Feb 21 ...	Microsoft Azure Sponsorship (0827c45d-52f6-48...	



* Subscription: Microsoft Azure Sponsorship (0827c45d-52f6-48fe-8ed1-a4a113ac31e1)

Resource group: examplerg

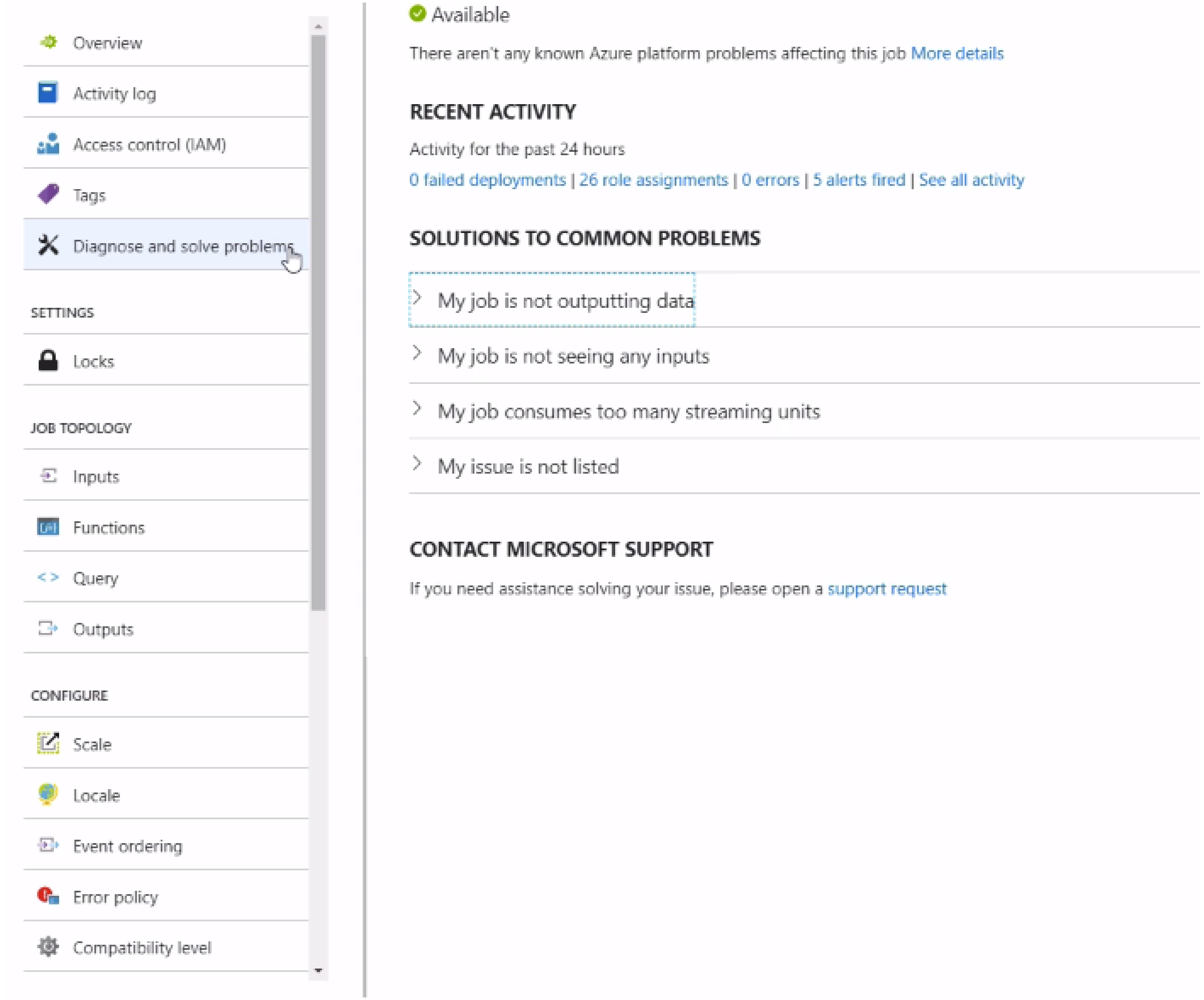
Resource type: Stream Analytics jobs

Resourc fraud-

Microsoft Azure Sponsorship (0827c45d-52f6-48fe-8ed1-a4a113ac31e1) > examplerg > fraud-detection

Turn on diagnostics to collect the following data.

- Execution
- Authoring
- AllMetrics



✓ Available

There aren't any known Azure platform problems affecting this job More details

RECENT ACTIVITY

Activity for the past 24 hours

0 failed deployments | 26 role assignments | 0 errors | 5 alerts fired | See all activity

SOLUTIONS TO COMMON PROBLEMS

- My job is not outputting data
- My job is not seeing any inputs
- My job consumes too many streaming units
- My issue is not listed

CONTACT MICROSOFT SUPPORT

If you need assistance solving your issue, please open a support request