DP-900: Microsoft Azure Data Fundamentals

MODULE 3

Module 3: Non-Relation Data in Azure

1

Explore Non-Relational Data Services in Azure

2

Provisioning Non-Relational Services in Azure

3

Managing Non-Relational data in Azure

Explore Non-relational data services in Azure

Explore use-cases and management benefits of using Azure Table storage

Explore use-cases and management benefits of using Azure Blob storage

Explore use-cases and management benefits of using Azure File storage

Explore use-cases and management benefits of using Azure Cosmos DB

Non-Relational Data Characteristics

```
## Customer 1 ID: 1
Name: Mark Hanson
Telephone: [ Home: 1-999-9999999, Business: 1-888-8888888, Cell: 1-777- 7777777 ]
Address: [ Home: 121 Main Street, Some City, NY, 10110,
           Business: 87 Big Building, Some City, NY, 10111 ]
## Customer 2 ID: 2
Title: Mr
Name: Jeff Hay
Telephone: [ Home: 0044-1999-333333, Mobile: 0044-17545-444444 ]
Address: [ UK: 86 High Street, Some Town, A County, GL8888, UK,
           US: 777 7th Street, Another City, CA, 90111 ]
```

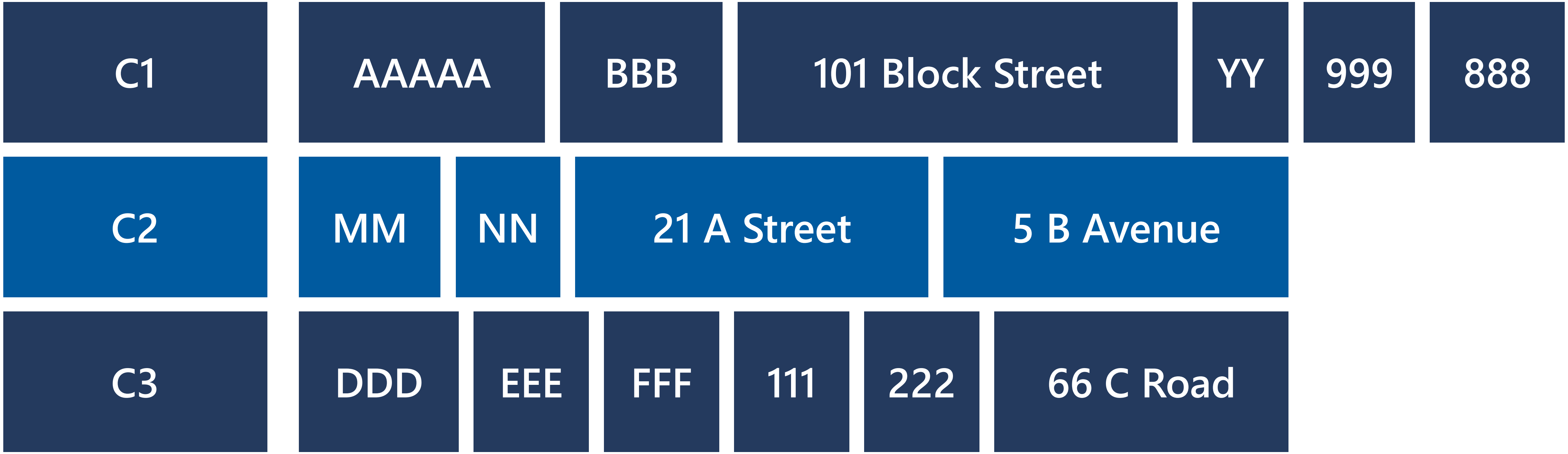
Non-relational collections can have:

Multiple entities in the same collection or container with different fields

Have a different, non-tabular schema

Are often defined by labeling each field with the name it represents

Azure Table Storage



Key (Customer ID)	Value (Customer Data)					
C1	AAAAA	BBB	101 Block Street	YY	999	888
C2	MM	NN	21 A Street	5 B Avenue		
C3	DDD	EEE	FFF	111	222	66 C Road

Azure Blob Storage

Block blobs

Has a maximum size of 4.7TB

Best for storing large, discrete, binary objects that changes infrequently

Each individual block can store up to 100MB of data

A block blob can contain up to 50000 blocks

Page blobs

Can hold up to 8TB of data

Is organized as a collection of fixed sized-512 byte pages

Used to implement virtual disk storage for virtual machines

Append blobs

The maximum size is just over 195GB

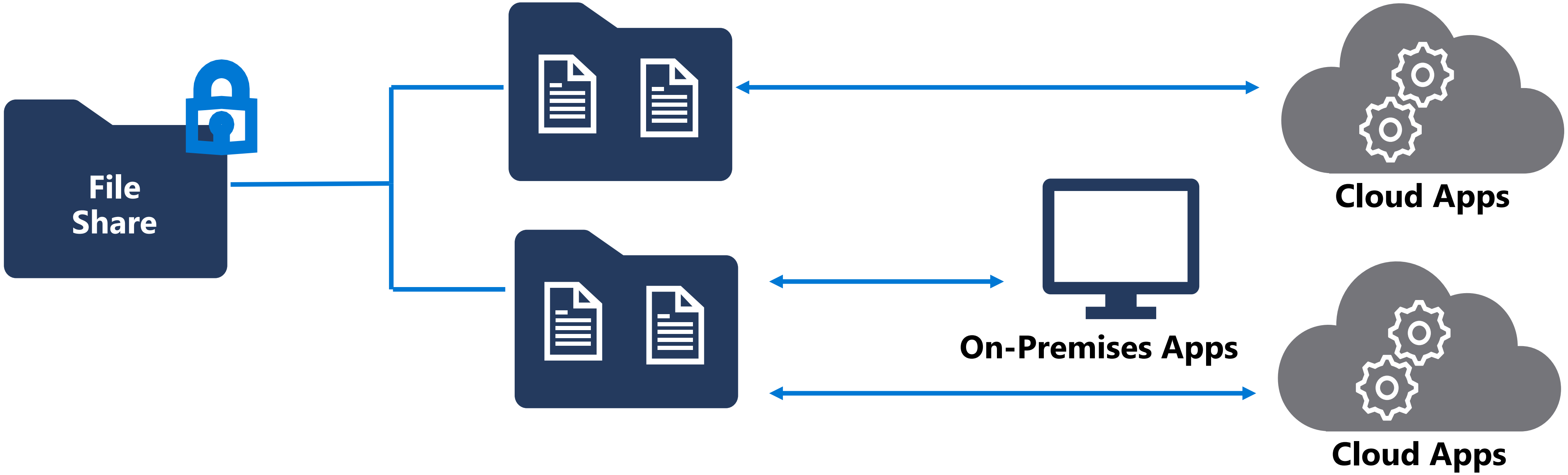
Is a block blob that is used to optimize append operations

Each individual block can store up to 4MB of data

Azure File Storage

Azure File Storage creates files shares in the cloud and provides the ability to access the file shares from anywhere with an internet connection.

- Uses Server Message Block 3.0 (SMB) to share files
- Share up to **100 TB** of data in a single storage account
- Fully managed service – data is replicated locally and is encrypted at rest



Azure Cosmos DB

Azure Cosmos DB is a multi-model NoSQL database management system.

Scalability

Performance

Availability

Programming model

- Cosmos DB manages data as a partitioned set of documents.
- Real time access with fast read and write latencies.
- Takes advantage of Azure scaling and storage capabilities.

Use cases for Azure Cosmos DB

Web and retail

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

Gaming

The database tier is a crucial component of gaming applications. Modern games perform graphical processing on mobile/console clients but rely on the cloud to deliver customized and personalized content like in-game stats, social media integration, and high-score leader boards

IoT scenarios

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

Lesson 1: Knowledge check (continued on next slide)

What are the elements of an Azure Table storage key?

☐ Table name and column name

☑ Partition key and row key

☐ Row number

When should you use a block blob, and when should you use a page blob?

☐ Use a block blob for unstructured data that requires random access to perform reads and writes. Use a page blob for discrete objects that rarely change.

☐ Use a block blob for active data stored using the Hot data access tier, and a page blob for data stored using the Cool or Archive data access tiers

☑ Use a page block for blobs that require random read and write access. Use a block blob for discrete objects that change infrequently

Why you might you use Azure File storage?

☐ To share files that are stored on-premises with users located at other sites

☑ To enable users at different sites to share files

☐ To store large binary data files containing images or other unstructured data

Lesson 1: Knowledge check (continued)

You are building a system that monitors the temperature throughout a set of office blocks, and sets the air conditioning in each room in each block to maintain a pleasant ambient temperature. Your system has to manage the air conditioning in several thousand buildings spread across the country/region, and each building typically contains at least 100 air-conditioned rooms. What type of NoSQL data store is most appropriate for capturing the temperature data to enable it to be processed quickly?

☑ Send the data to an Azure Cosmos DB database and use Azure Functions to process the data

□ Store the data in a file stored in a share created using Azure File Storage

□ Write the temperatures to a blob in Azure Blob storage

Module 3: Non-Relation Data in Azure

1

Explore Non-Relational Data Services in Azure

2

Provisioning Non-Relational Services in Azure

3

Managing Non-Relational data in Azure

Explore provisioning and deploying non-relational data services in Azure

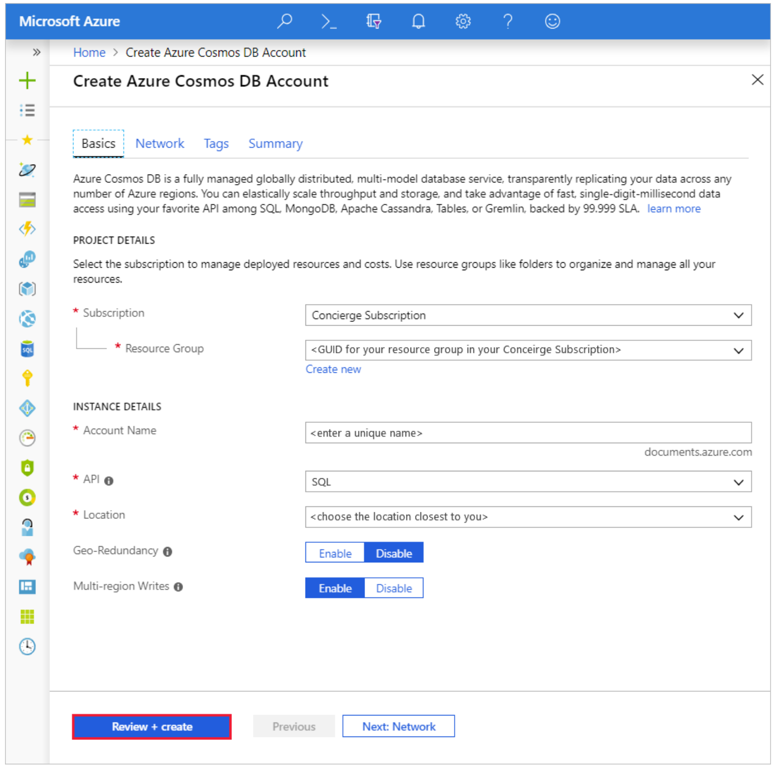
Provision non-relational data services

Configure non-relational data services

Explore basic connectivity issues

Explore data security components

Provisioning Cosmos DB



Create Azure Cosmos DB Account

Azure Cosmos DB is a fully managed globally distributed, multi-model database service, transparently replicating your data across any number of Azure regions. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access using your favorite API among SQL, MongoDB, Apache Cassandra, Tables, or Gremlin, backed by 99.999 SLA. 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: Concierge Subscription

* Resource Group: <GUID for your resource group in your Conceirge Subscription>

INSTANCE DETAILS

* Account Name: <enter a unique name> documents.azure.com

* API: SQL

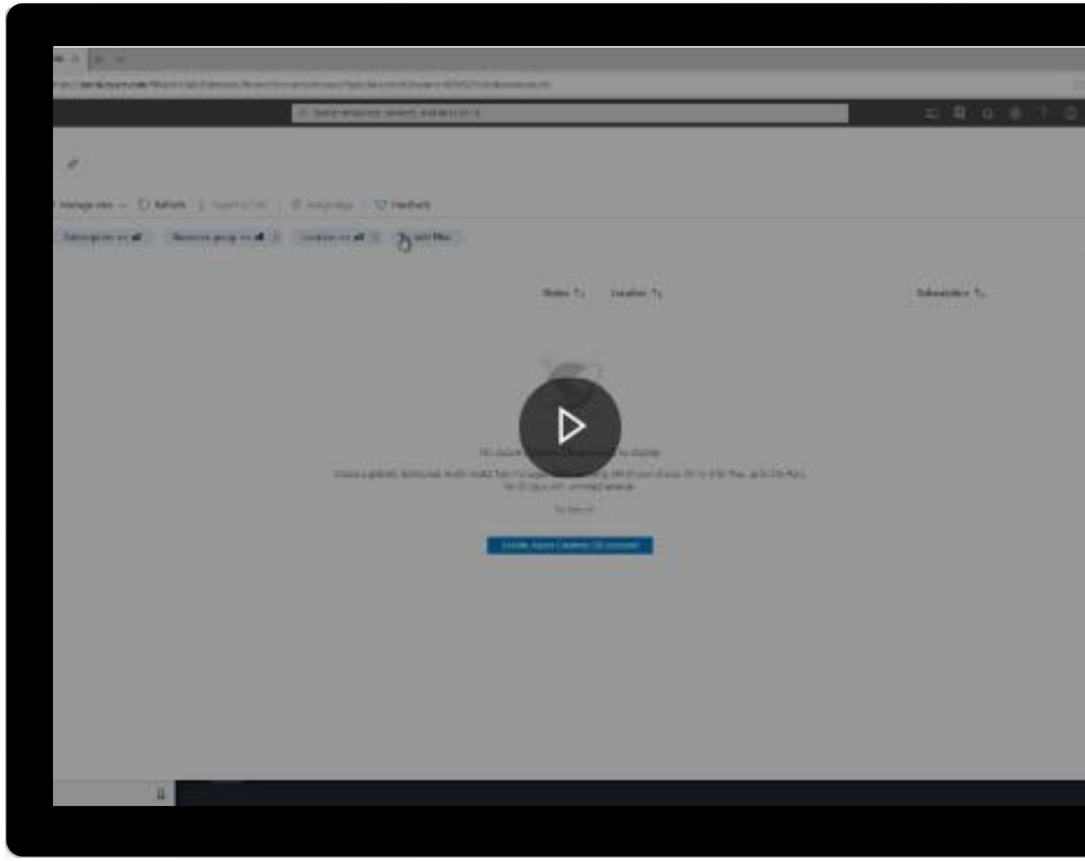
* Location: <choose the location closest to you>

Geo-Redundancy: Enable [✓ Disable]

Multi-region Writes: [✓ Enable] Disable

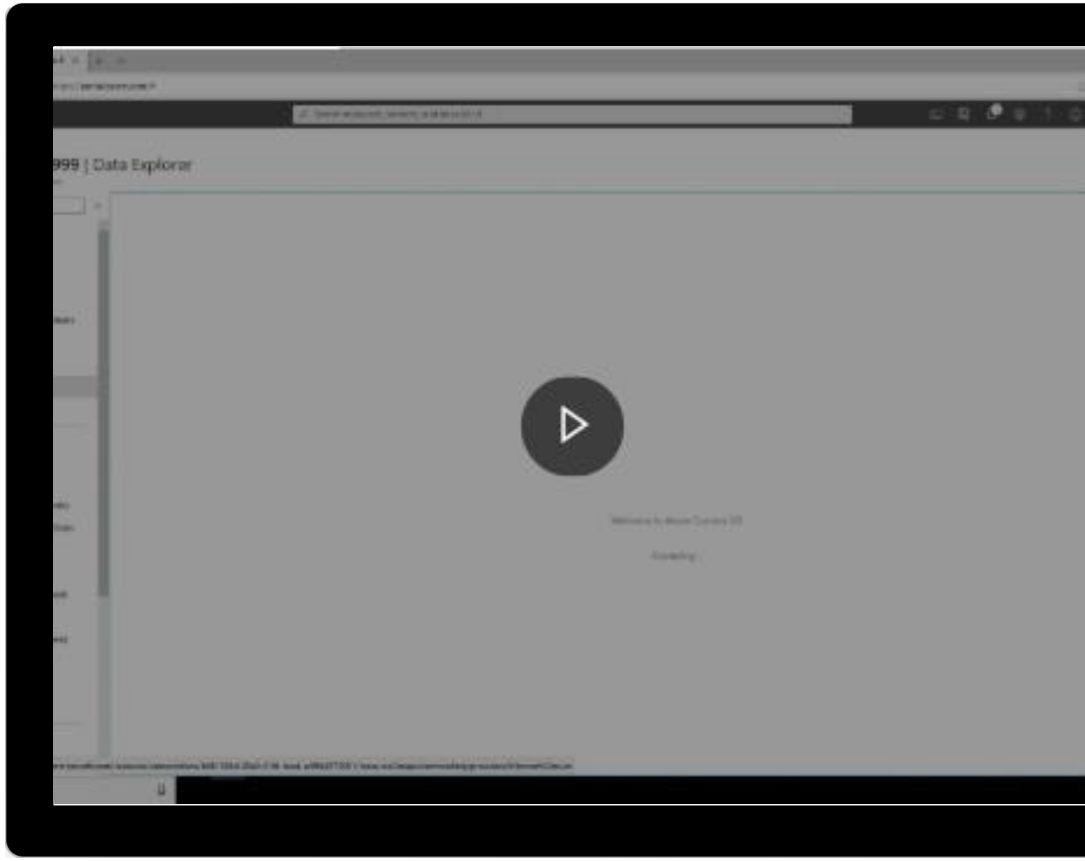
Demo: How to provision a Cosmos DB account

You can provision a Cosmos DB account interactively using the Azure portal, or you can perform this task programmatically through the Azure CLI, Azure PowerShell, or an Azure Resource Manager template. This video describes how to use the Azure portal

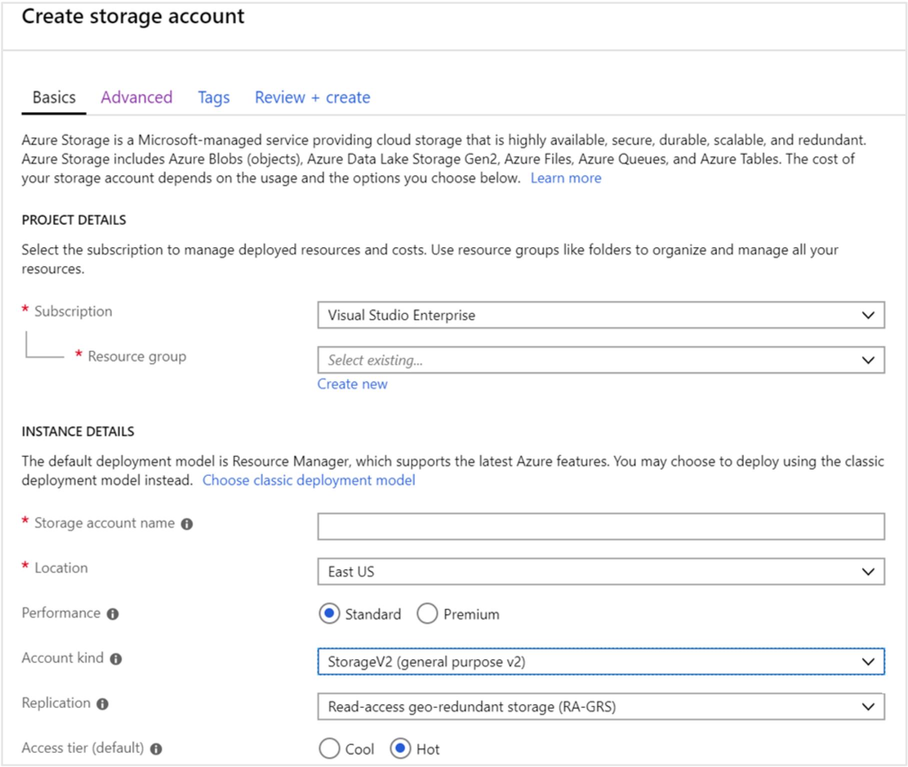


Demo

Use the Azure portal to create a database and container



Provisioning Data Lake storage



Create storage account

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

PROJECT DETAILS

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

* Subscription: Visual Studio Enterprise

* Resource group: Select existing...

Create new

INSTANCE DETAILS

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

* Storage account name:

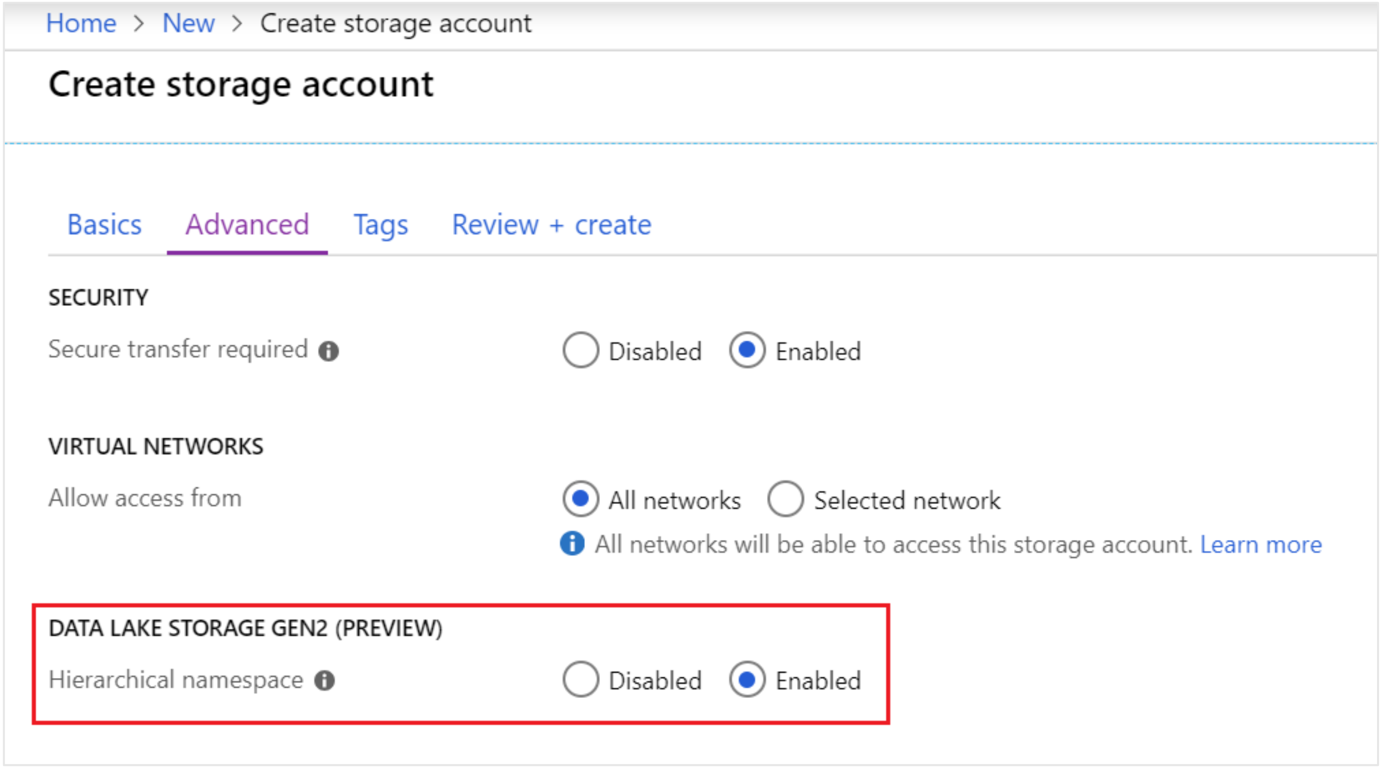
* Location: East US

Performance: ● Standard ○ Premium

Account kind: StorageV2 (general purpose v2)

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

Access tier (default): ○ Cool ● Hot



Create storage account

SECURITY

Secure transfer required: ○ Disabled ● Enabled

VIRTUAL NETWORKS

Allow access from: ● All networks ○ Selected network

All networks will be able to access this storage account. Learn more

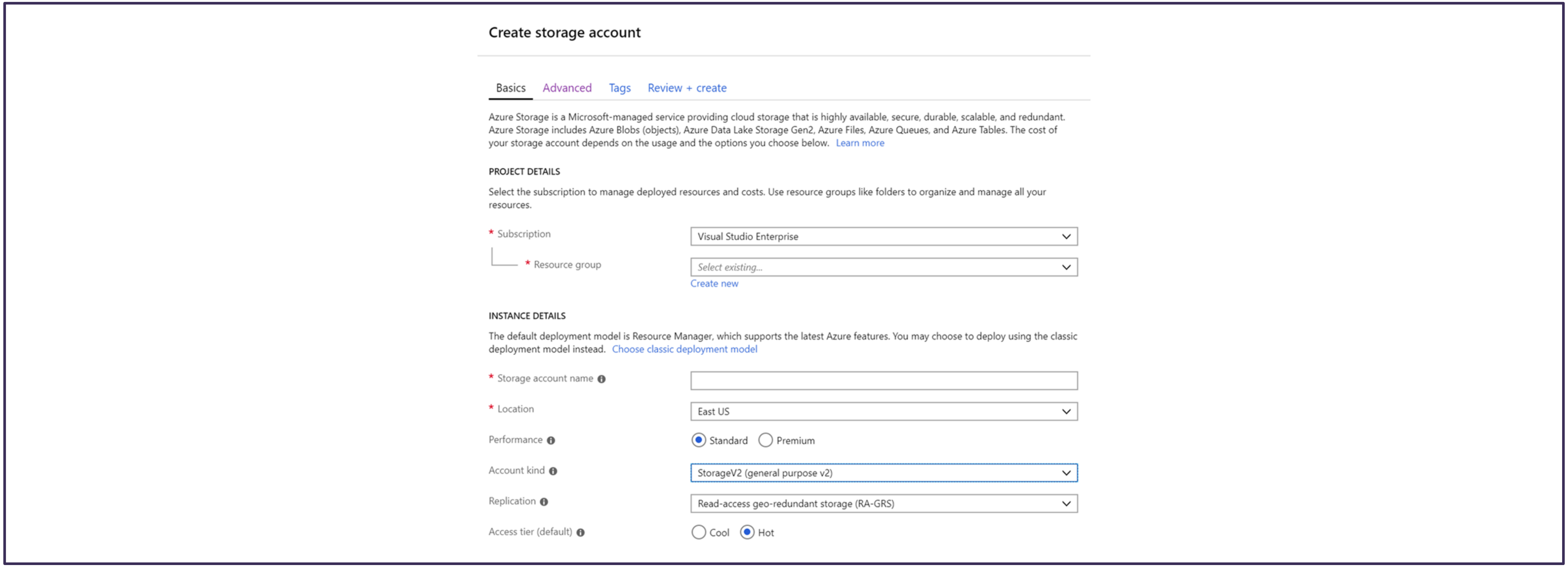
DATA LAKE STORAGE GEN2 (PREVIEW)

Hierarchical namespace: ○ Disabled ● Enabled

Demo: Azure authentication

Azure AD is a separate Azure service. You add users and other security principals (such as an application) to a security domain managed by Azure AD. This video describes how authentication works with Azure

Configure storage accounts



Create storage account

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

PROJECT DETAILS

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

* Subscription	Visual Studio Enterprise
* Resource group	Select existing... Create new

INSTANCE DETAILS

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

* Storage account name	
* Location	East US
Performance	● Standard ○ Premium
Account kind	StorageV2 (general purpose v2)
Replication	Read-access geo-redundant storage (RA-GRS)
Access tier (default)	○ Cool ● Hot

Lesson 2: Knowledge check

What is provisioning?

- ☑ The act of running series of tasks that a service provider performs to create and configure a service
- □ Providing other users access to an existing service
- □ Tuning a service to improve performance

What is a security principal?

- □ A named collection of permissions that can be granted to a service, such as the ability to use the service to read, write, and delete data. In Azure, examples include **Owner** and **Contributor.**
- □ A set of resources managed by a service to which you can grant access
- ☑ An object that represents a user, group, service, or managed identity that is requesting access to Azure resources

Which of the following is an advantage of using multi-region replication with Cosmos DB?

- □ Data will always be consistent in every region
- ☑ Availability is increased
- □ Increased security for your data

Module 3: Non-Relation Data in Azure

1

Explore Non-Relational Data Services in Azure

2

Provisioning Non-Relational Services in Azure

3

Managing Non-Relational data in Azure

Explore provisioning and deploying non-relational data services in Azure

- Upload data to a Cosmos DB database, and learn how to query this data

- Upload and download data in an Azure Storage account

Cosmos DB APIs

SQL API

- Supports SQL-like query language

Table API

- Compatible with Azure Table Storage

MongoDB API

- Compatible with MongoDB

Cassandra API

- Compatible with Cassandra

Gremlin API

- A graph database

Load data using the Cosmos DB Migration tool

You can use the Data Migration tool to import data to Azure Cosmos DB from a variety of sources, including:

JSON files

MongoDB

SQL Server

CSV files

Azure Table storage

Amazon DynamoDB

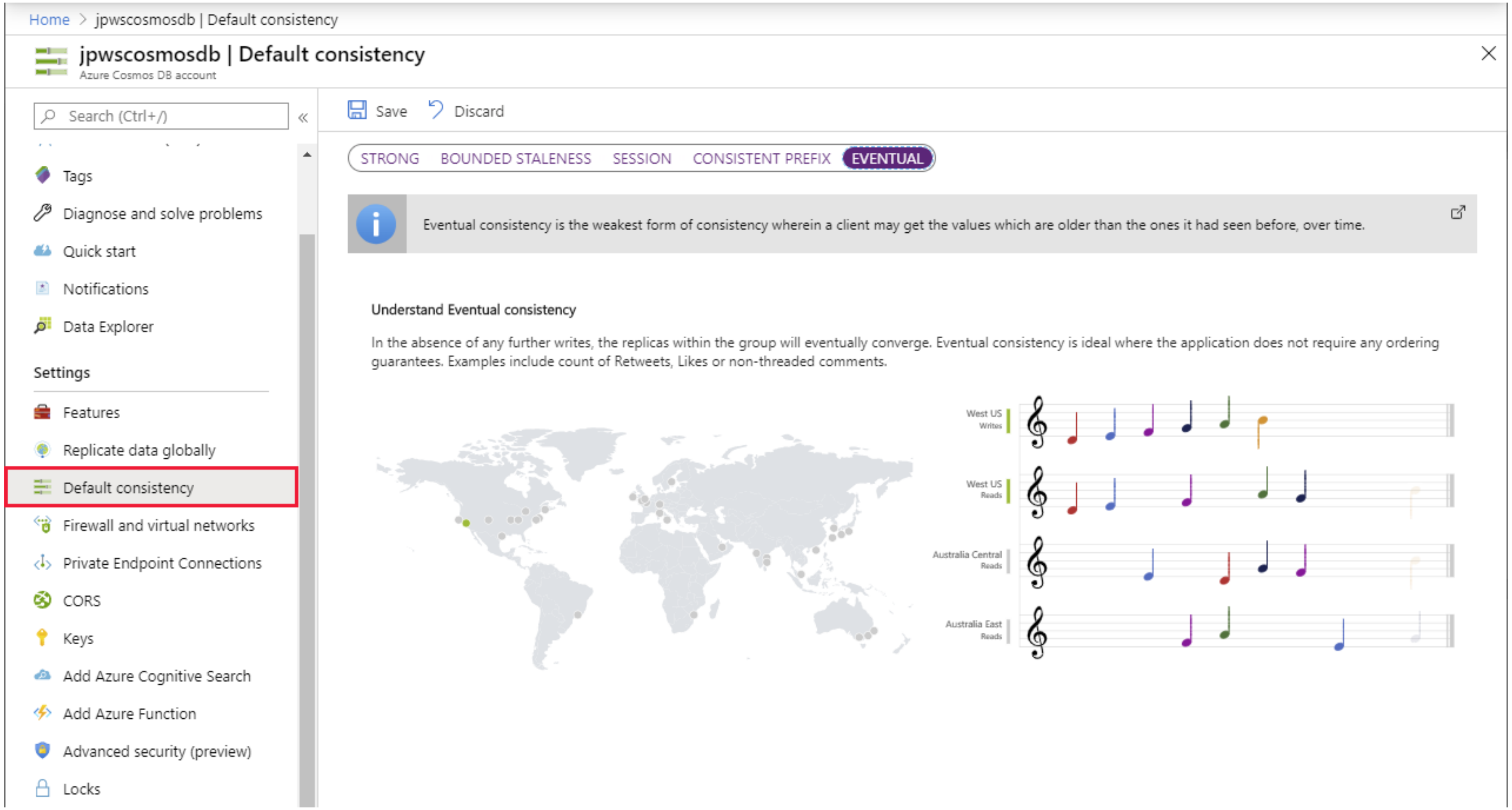
HBase

Azure Cosmos containers

Demo: Configure consistency

Within a single region, Cosmos DB uses a cluster of servers. This approach helps to improve scalability and availability. A copy of all data is held in each server in the cluster. This video explains how this works, and the effects it can have on consistency

Consistency in Azure Cosmos DB

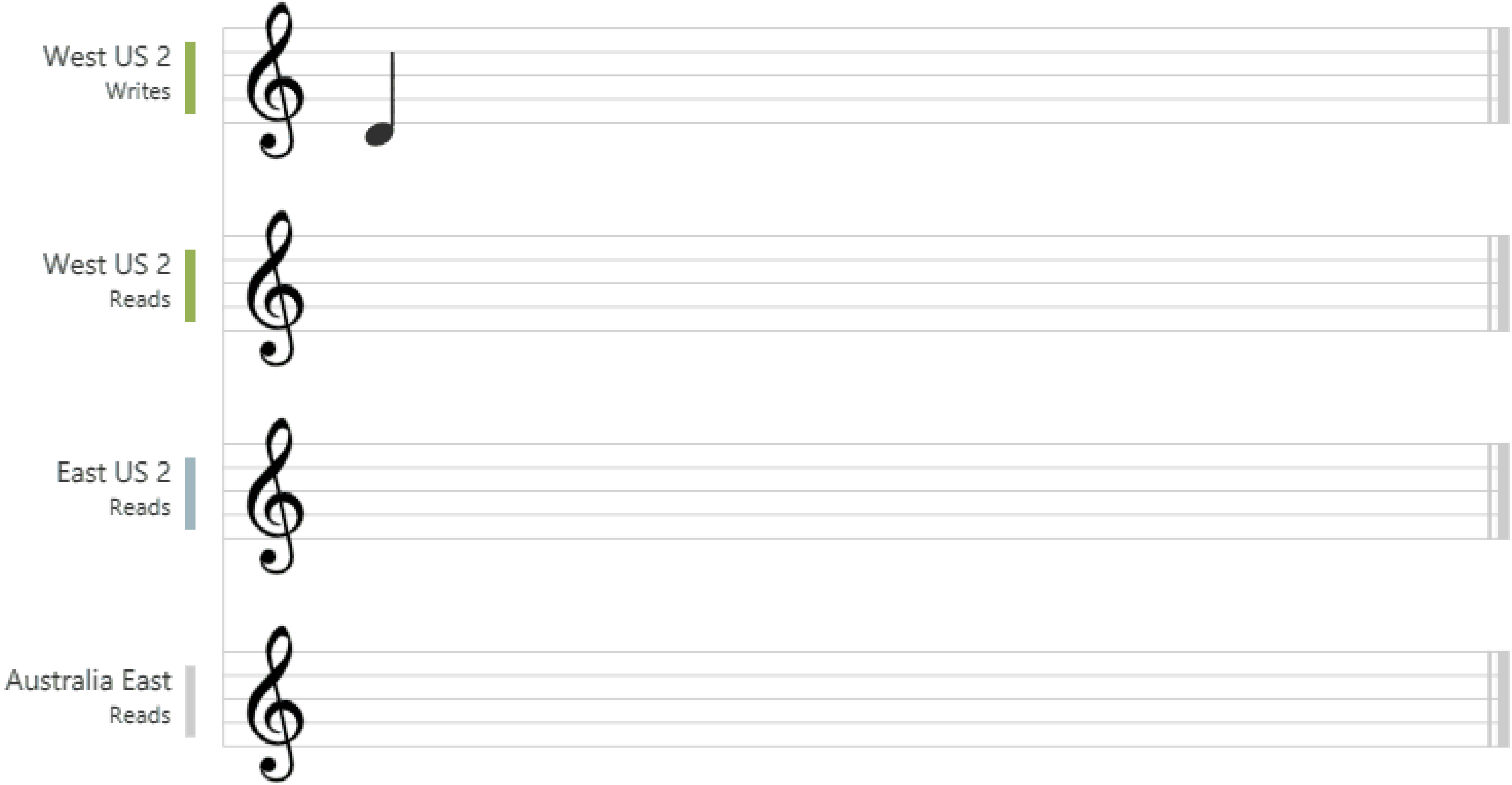


Eventual consistency is the weakest form of consistency wherein a client may get the values which are older than the ones it had seen before, over time.

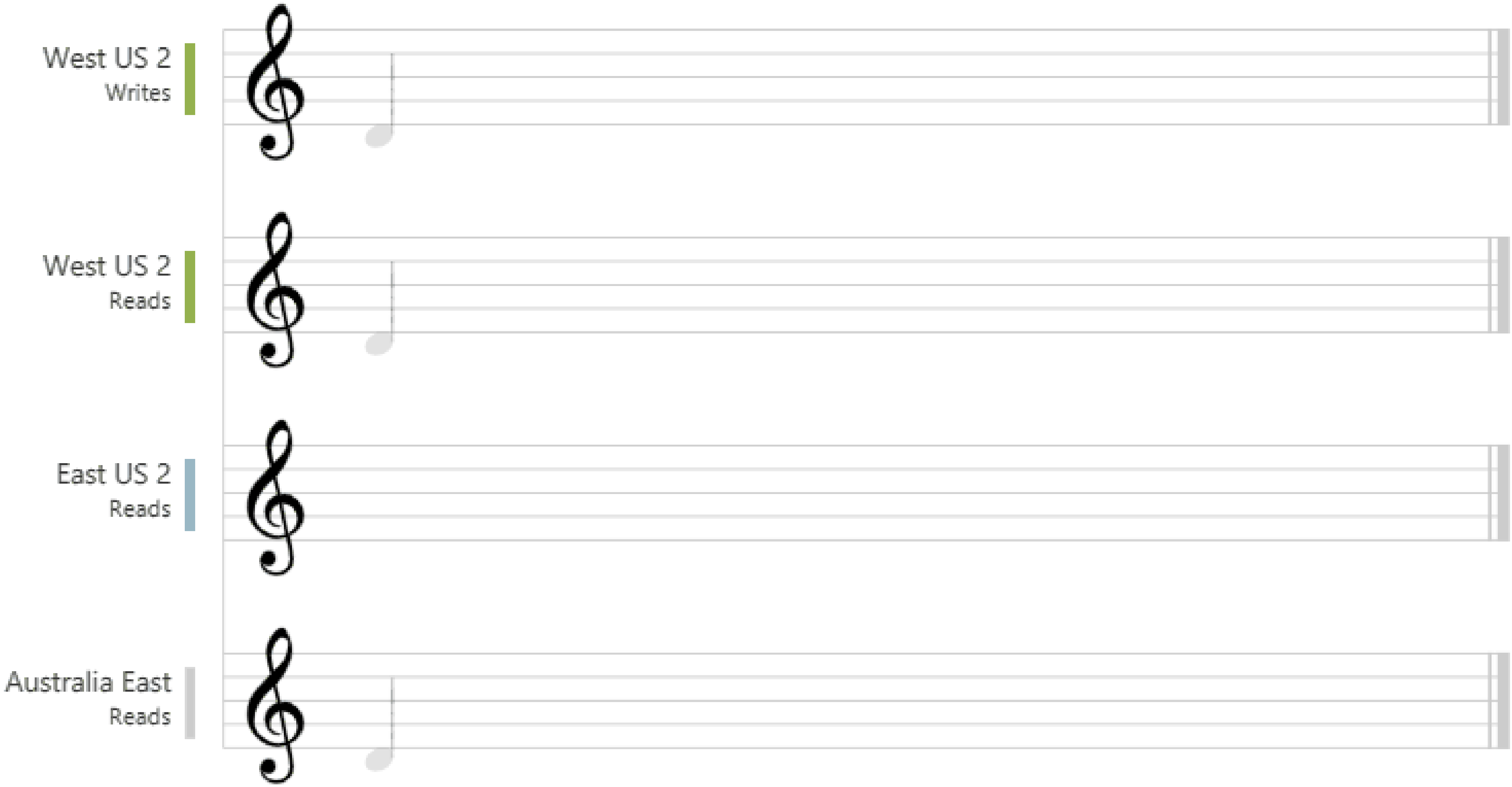
Understand Eventual consistency

In the absence of any further writes, the replicas within the group will eventually converge. Eventual consistency is ideal where the application does not require any ordering guarantees. Examples include count of Retweets, Likes or non-threaded comments.

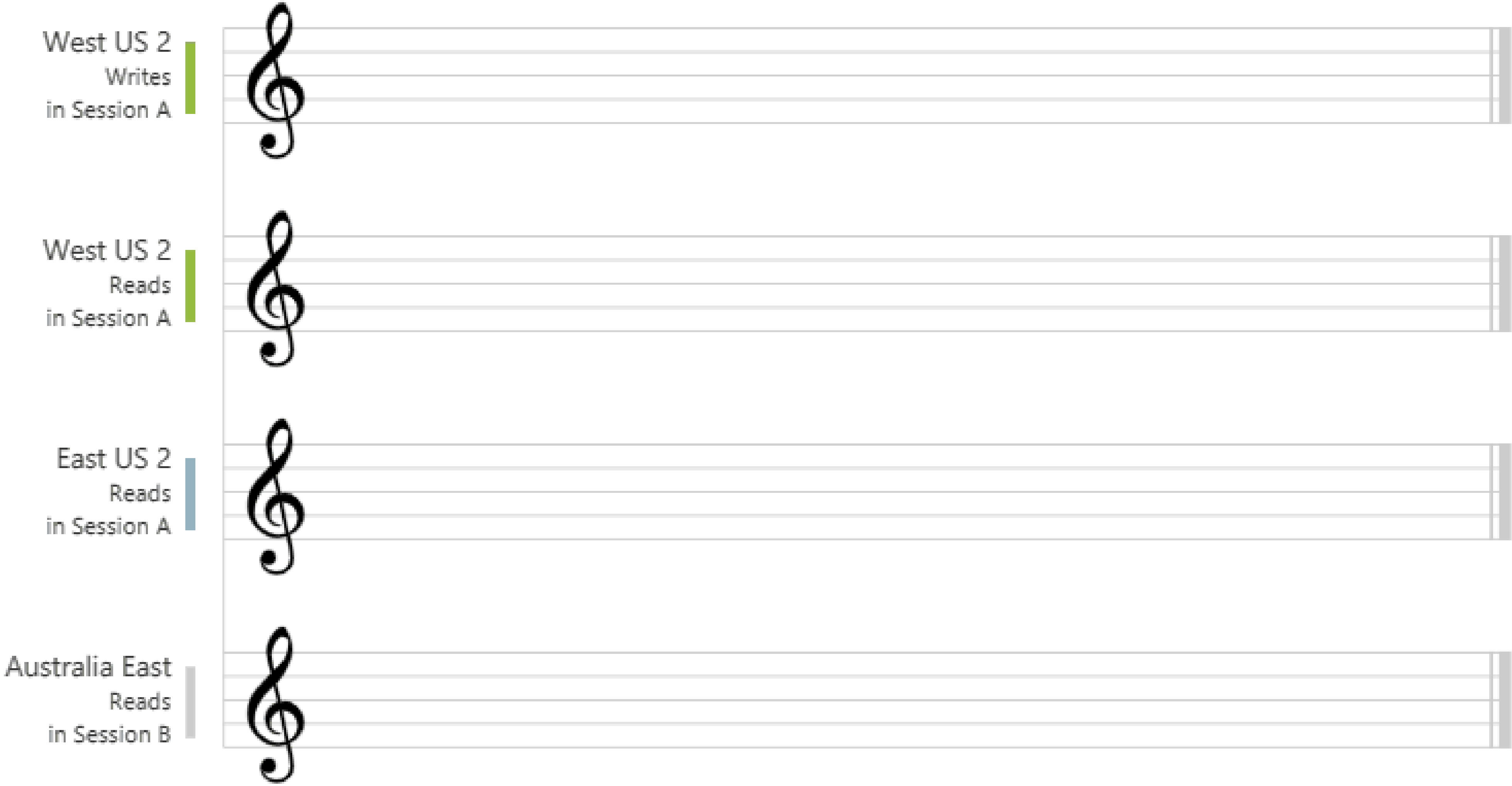
Eventual consistency



Consistent prefix consistency



Session consistency



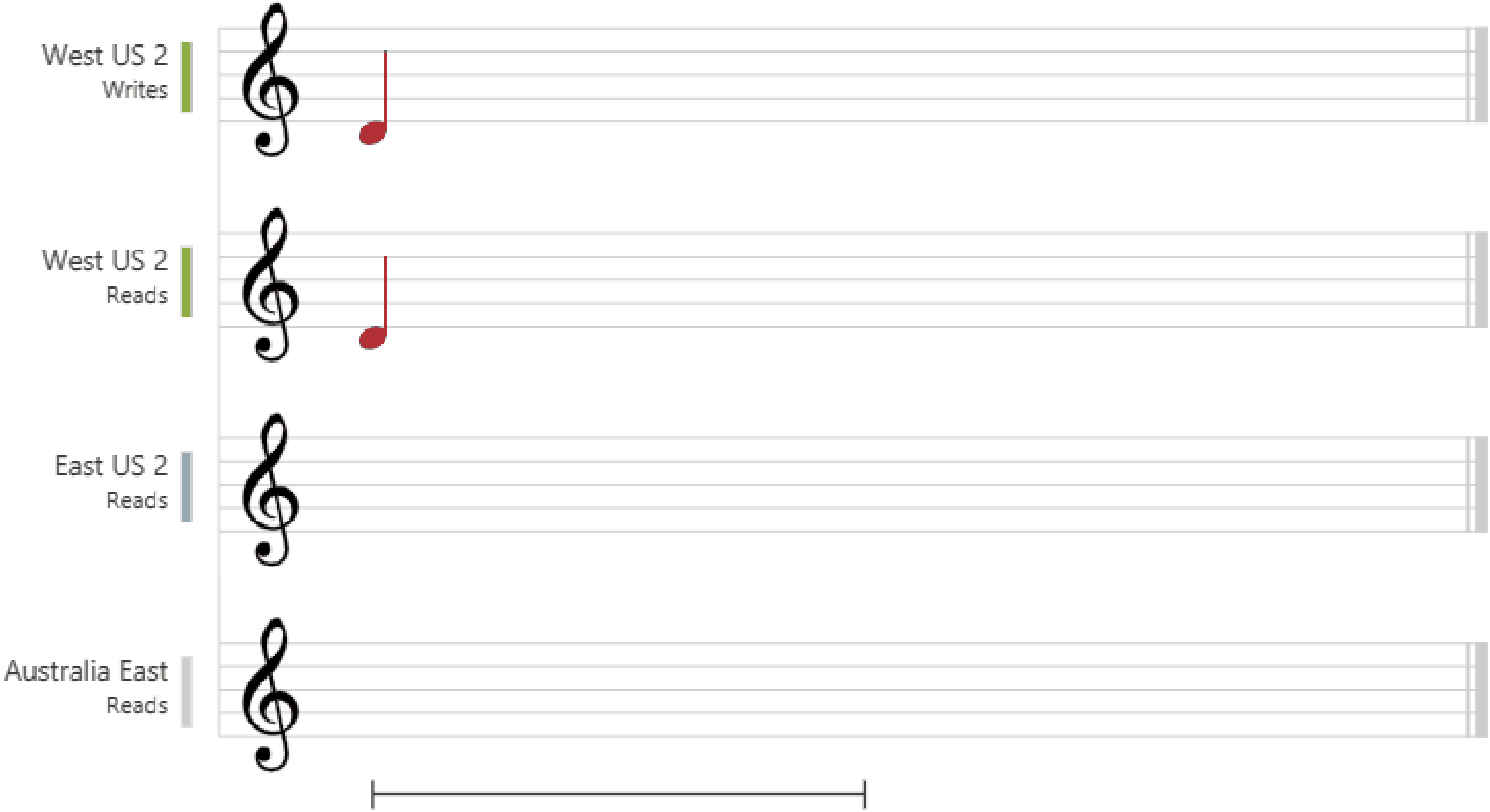
West US 2
Writes
in Session A

West US 2
Reads
in Session A

East US 2
Reads
in Session A

Australia East
Reads
in Session B

Bounded staleness consistency



Maximum delay of 100 versions (K) of a record or 5 seconds, whichever is smaller

Strong Consistency

Cosmos DB SQL API

Aggregation Function Basics

```
COUNT( <fields_to_count> )
SUM( <numeric_fields> )
AVG( <numeric_fields> )
MAX( <numeric_fields> )
MIN( <numeric_fields> )
```

SQL API examples

```
SELECT COUNT(*) FROM Products p

SELECT SUM(p.quantity) FROM Products p
WHERE p.expired = 0

SELECT AVG(p.price) AS ‘Average Price’
FROM Products p

SELECT p1.ID, p.Name, p1.Description,
p1.Price FROM Products p1
WHERE p1.Price = (SELECT MIN(p2.Price) FROM
Product p2)
```

Lab: Upload, download, and query data in a non-relational data store

Go to the exercise **Upload, download, and query data in a non-relational data store** module on Microsoft Learn, and follow the instructions in the module