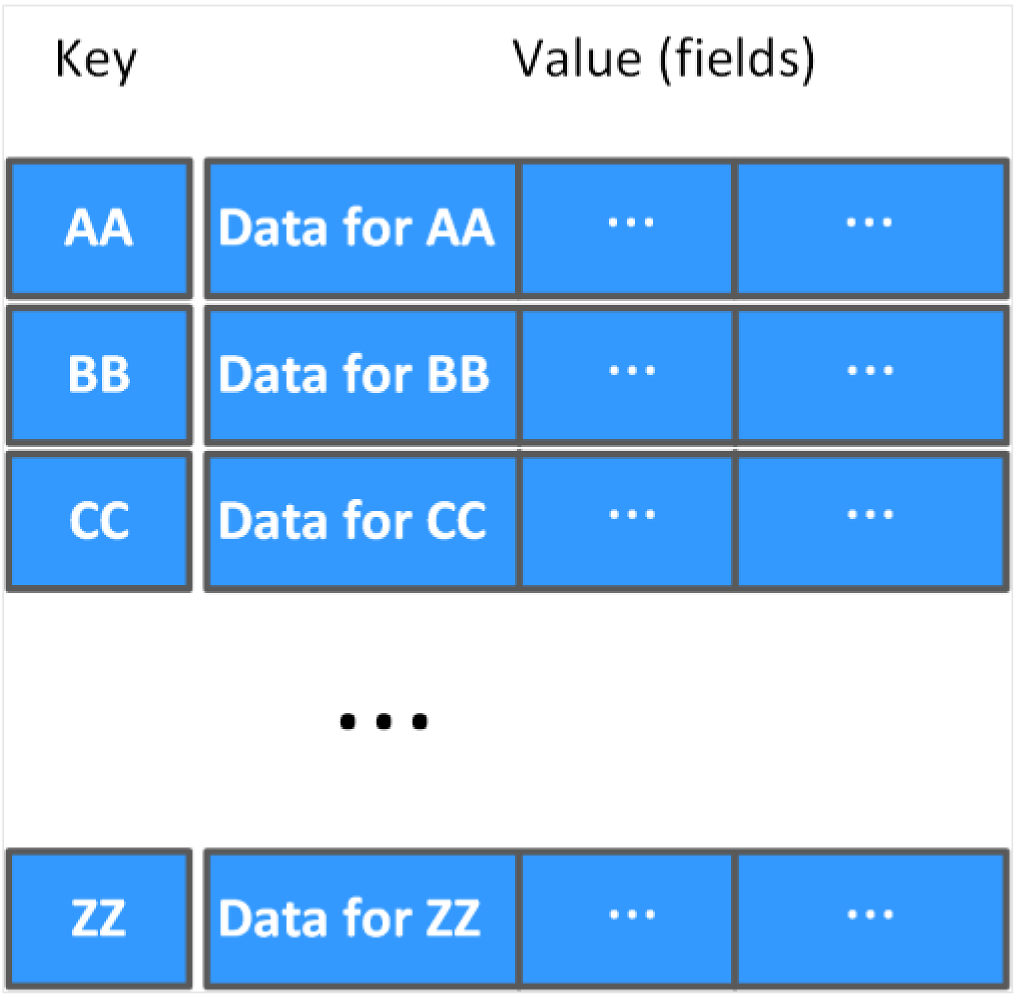
✓ 100 XP

Explore Azure Table storage

12 minutes

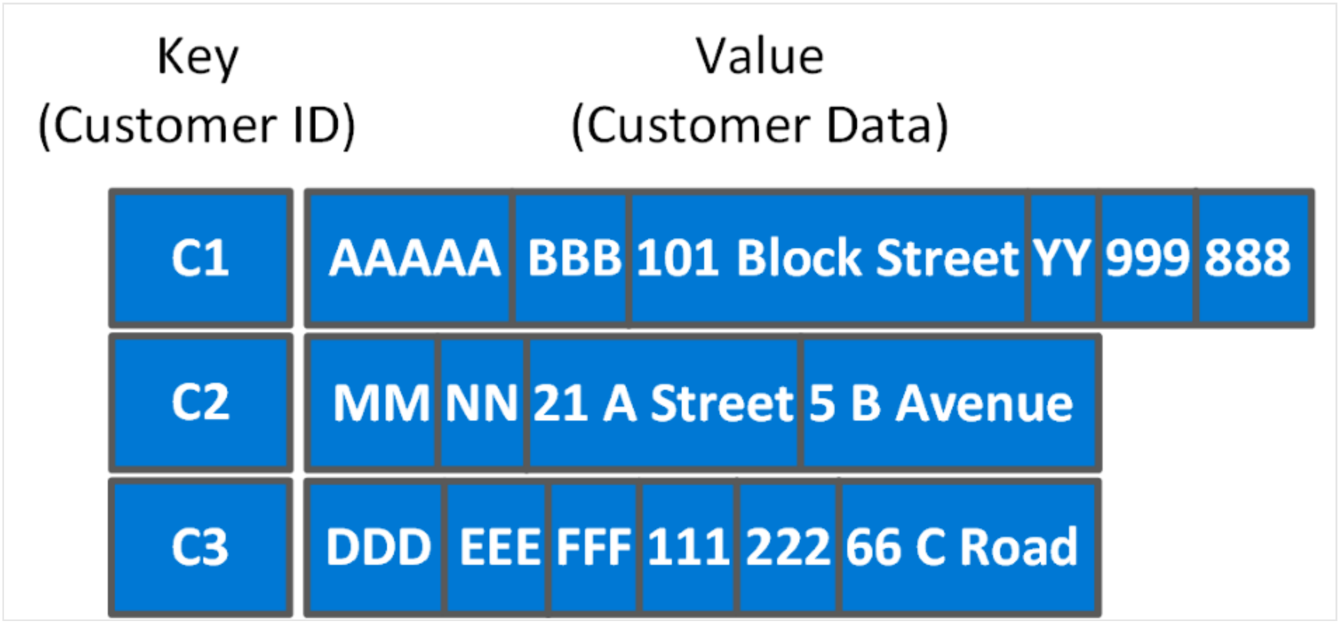
Azure Table Storage implements the NoSQL key-value model. In this model, the data for an item is stored as a set of fields, and the item is identified by a unique key.



What is Azure Table Storage?

Azure Table Storage is a scalable key-value store held in the cloud. You create a table using an Azure storage account.

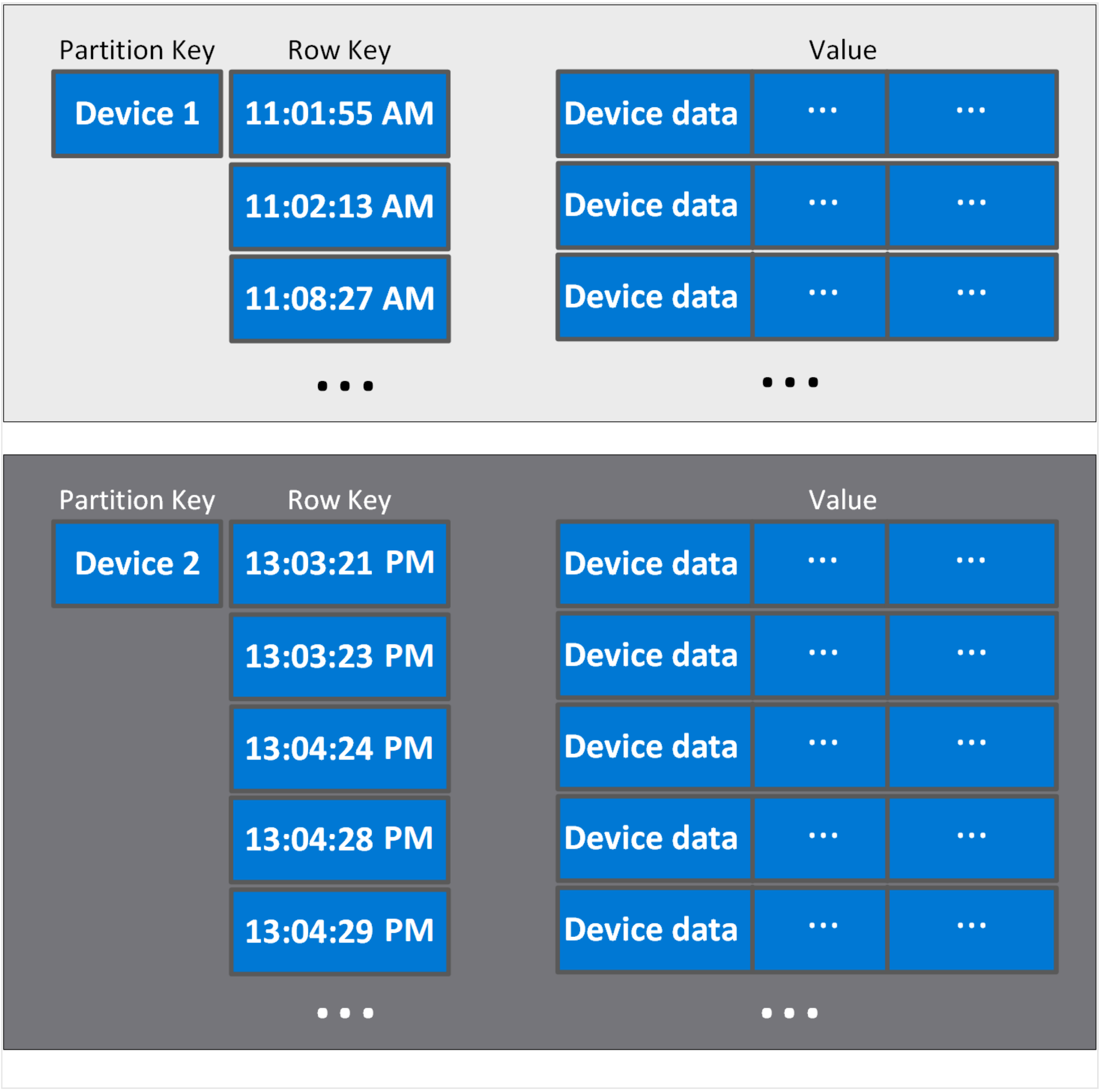
In an Azure Table Storage table, items are referred to as *rows*, and fields are known as *columns*. However, don't let this terminology confuse you by thinking that an Azure Table Storage table is like a table in a relational database. An Azure table enables you to store semi-structured data. All rows in a table must have a key, but apart from that the columns in each row can vary. Unlike traditional relational databases, Azure Table Storage tables have no concept of relationships, stored procedures, secondary indexes, or foreign keys. Data will usually be denormalized, with each row holding the entire data for a logical entity. For example, a table holding customer information might store the first name, last name, one or more telephone numbers, and one or more addresses for each customer. The number of fields in each row can be different, depending on the number of telephone numbers and addresses for each customer, and the details recorded for each address. In a relational database, this information would be split across multiple rows in several tables. In this example, using Azure Table Storage provides much faster access to the details of a customer because the data is available in a single row, without requiring that you perform joins across relationships.



To help ensure fast access, Azure Table Storage splits a table into partitions. Partitioning is a mechanism for grouping related rows, based on a common property or *partition key*. Rows that share the same partition key will be stored together. Partitioning not only helps to organize data, it can also improve scalability and performance:

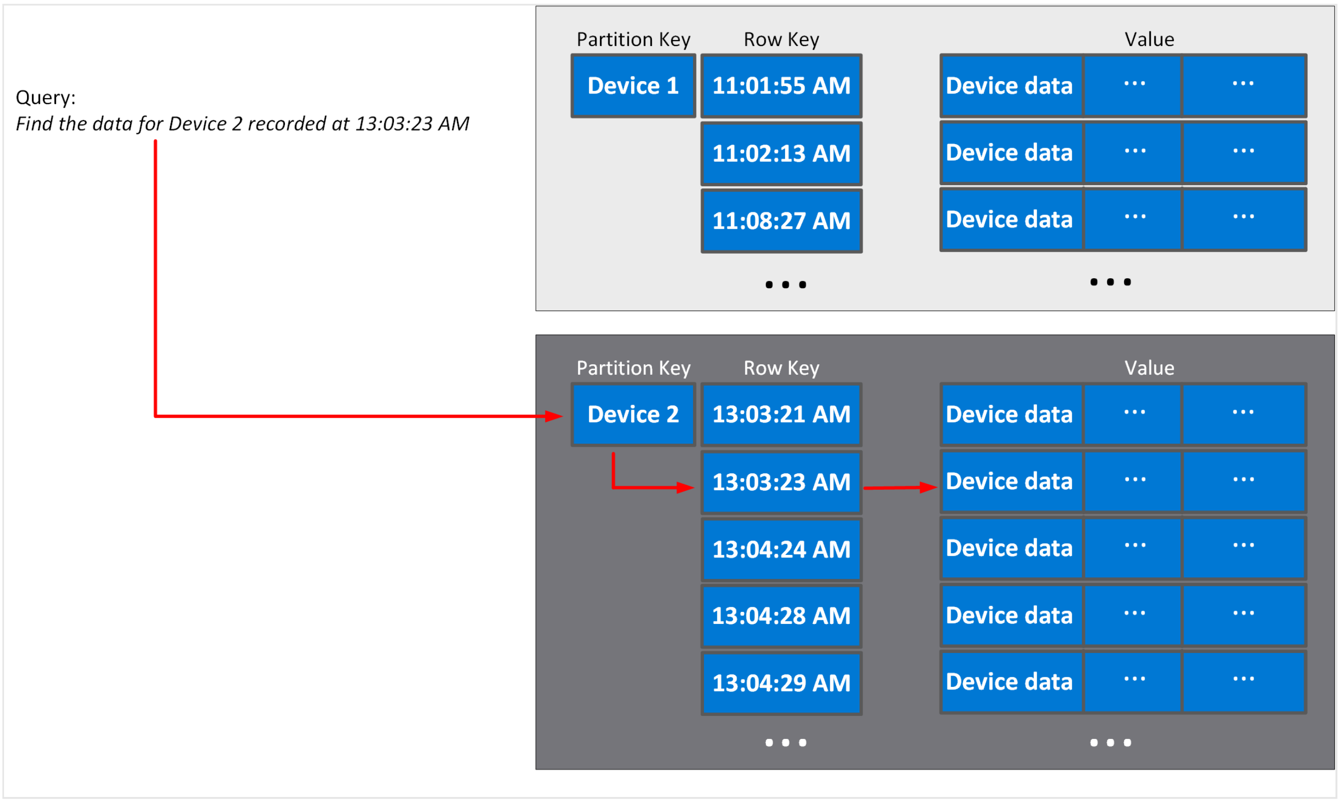
- Partitions are independent from each other, and can grow or shrink as rows are added to, or removed from, a partition. A table can contain any number of partitions.
- When you search for data, you can include the partition key in the search criteria. This helps to narrow down the volume of data to be examined, and improves performance by reducing the amount of I/O (reads and writes) needed to locate the data.

The key in an Azure Table Storage table comprises two elements; the partition key that identifies the partition containing the row (as described above), and a row key that is unique to each row in the same partition. Items in the same partition are stored in row key order. If an application adds a new row to a table, Azure ensures that the row is placed in the correct position in the table. In the example below, taken from an IoT scenario, the row key is a date and time value.

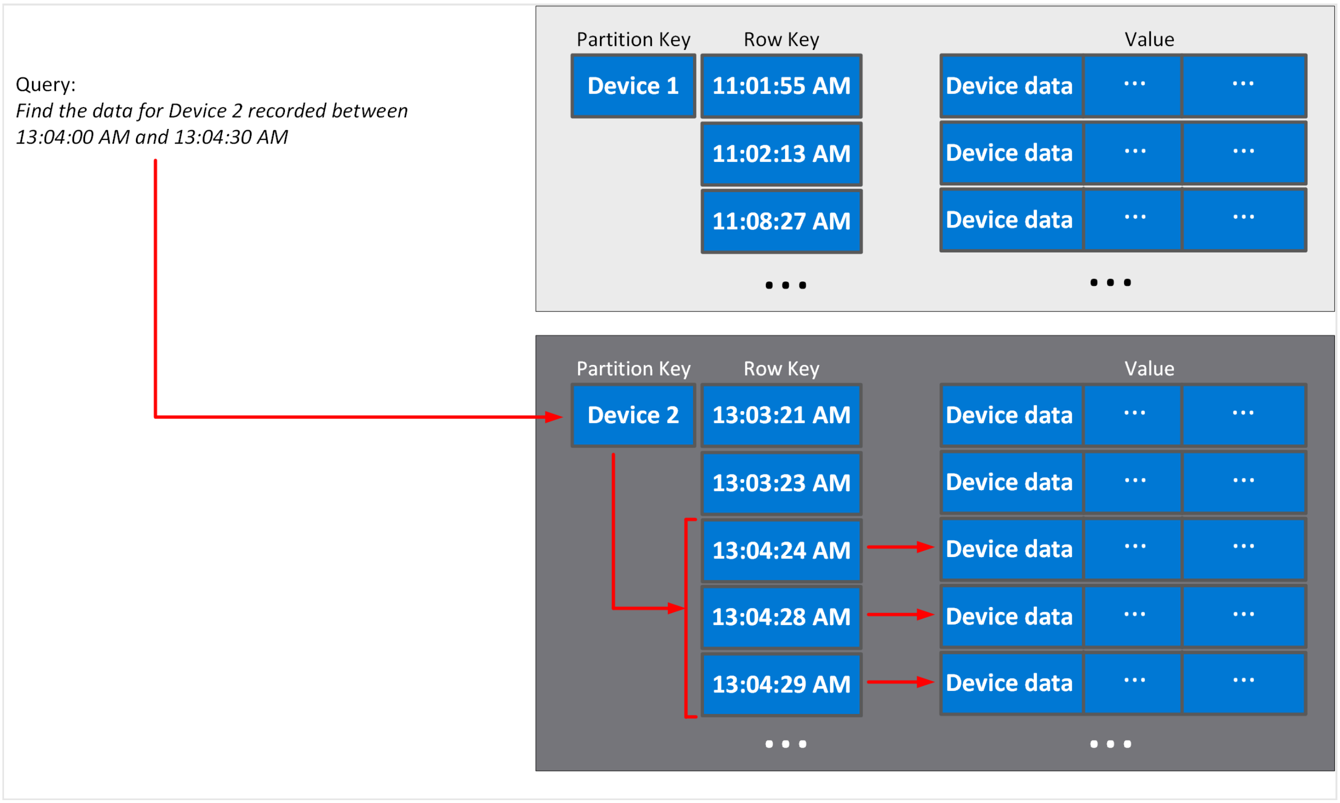


This scheme enables an application to quickly perform *Point queries* that identify a single row, and *Range queries* that fetch a contiguous block of rows in a partition.

In a point query, when an application retrieves a single row, the partition key enables Azure to quickly hone in on the correct partition, and the row key lets Azure identify the row in that partition. You might have hundreds of millions of rows, but if you've defined the partition and row keys carefully when you designed your application, data retrieval can be very quick. The partition key and row key effectively define a clustered index over the data.



In a range query, the application searches for a set of rows in a partition, specifying the start and end point of the set as row keys. This type of query is also very quick, as long as you have designed your row keys according to the requirements of the queries performed by your application.



The columns in a table can hold numeric, string, or binary data up to 64 KB in size. A table can have up to 252 columns, apart from the partition and row keys. The maximum row size is 1 MB. For more information, read Understanding the Table service data model.

Use cases and management benefits of using Azure Table Storage

Azure Table Storage tables are schemaless. It's easy to adapt your data as the needs of your application evolve. You can use tables to hold flexible datasets such as user data for web applications, address books, device information, or other types of metadata your service requires. The important part is to choose the partition and row keys carefully.

The primary advantages of using Azure Table Storage tables over other ways of storing data include:

- It's simpler to scale. It takes the same time to insert data in an empty table, or a table with billions of entries. An Azure storage account can hold up to 5 PB of data.
- A table can hold semi-structured data
- There's no need to map and maintain the complex relationships typically required by a normalized relational database.
- Row insertion is fast
- Data retrieval is fast, if you specify the partition and row keys as query criteria

There are disadvantages to storing data this way though, including:

- Consistency needs to be given consideration as transactional updates across multiple entities aren't guaranteed
- There's no referential integrity; any relationships between rows need to be maintained externally to the table
- It's difficult to filter and sort on non-key data. Queries that search based on non-key fields could result in full table scans

Azure Table Storage is an excellent mechanism for:

- Storing TBs of structured data capable of serving web scale applications. Examples include product catalogs for eCommerce applications, and customer information, where the data can be quickly identified and ordered by a composite key. In the case of a product catalog, the partition key could be the product category (such as footwear), and the row key identifies the specific product in that category (such as climbing boots).
- Storing datasets that don't require complex joins, foreign keys, or stored procedures, and that can be denormalized for fast access. In an IoT system, you might use Azure Table Storage to capture device sensor data. Each device could have its own partition, and the data could be ordered by the date and time each measurement was captured.
- Capturing event logging and performance monitoring data. Event log and performance information typically contain data that is structured according to the type of event or performance measure being recorded. The data could be partitioned by event or

performance measurement type, and ordered by the date and time it was recorded. Alternatively, you could partition data by date, if you need to analyze an ordered series of events and performance measures chronologically. If you want to analyze data by type and date/time, then consider storing the data twice, partitioned by type, and again by date. Writing data is fast, and the data is static once it has been recorded.

Azure Table Storage is intended to support very large volumes of data, up to several hundred TBs in size. As you add rows to a table, Azure Table Storage automatically manages the partitions in a table and allocates storage as necessary. You don't need to take any additional steps yourself.

Azure Table Storage provides high-availability guarantees in a single region. The data for each table is replicated three times within an Azure region. For increased availability, but at additional cost, you can create tables in geo-redundant storage. In this case, the data for each table is replicated a further three times in another region several hundred miles away. If a replica in the local region becomes unavailable, Azure will transparently switch to a working replica while the failed replica is recovered. If an entire region is hit by an outage, your tables are safe in a remote region, and you can quickly switch your application to connect to that remote region.

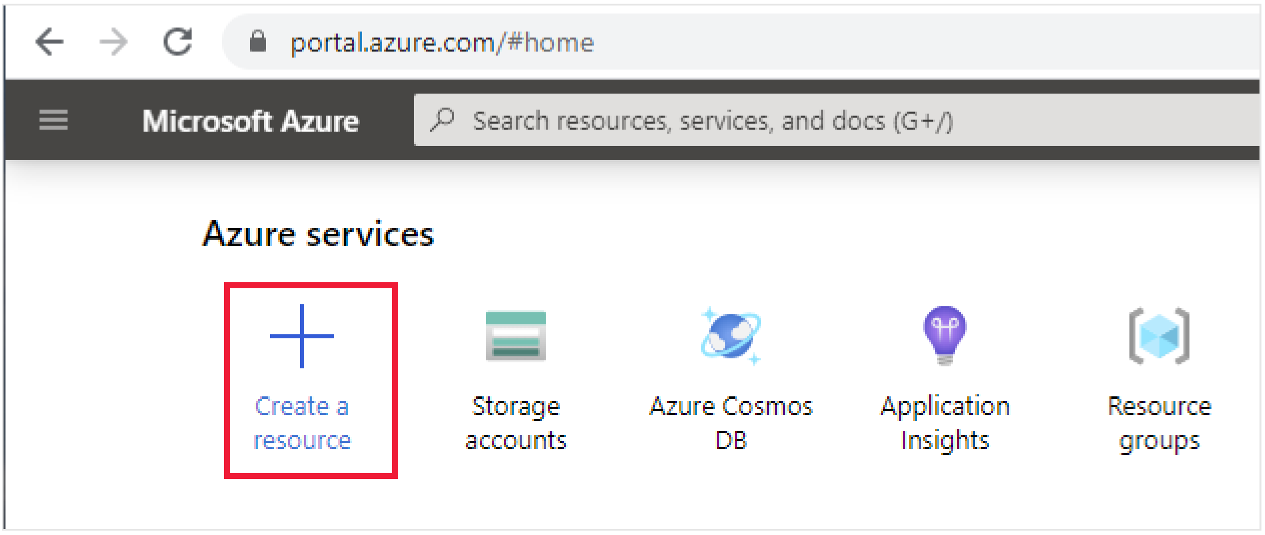
Azure Table Storage helps to protect your data. You can configure security and role-based access control to ensure that only the people or applications that need to see your data can actually retrieve it.

Create and view a table using the Azure portal

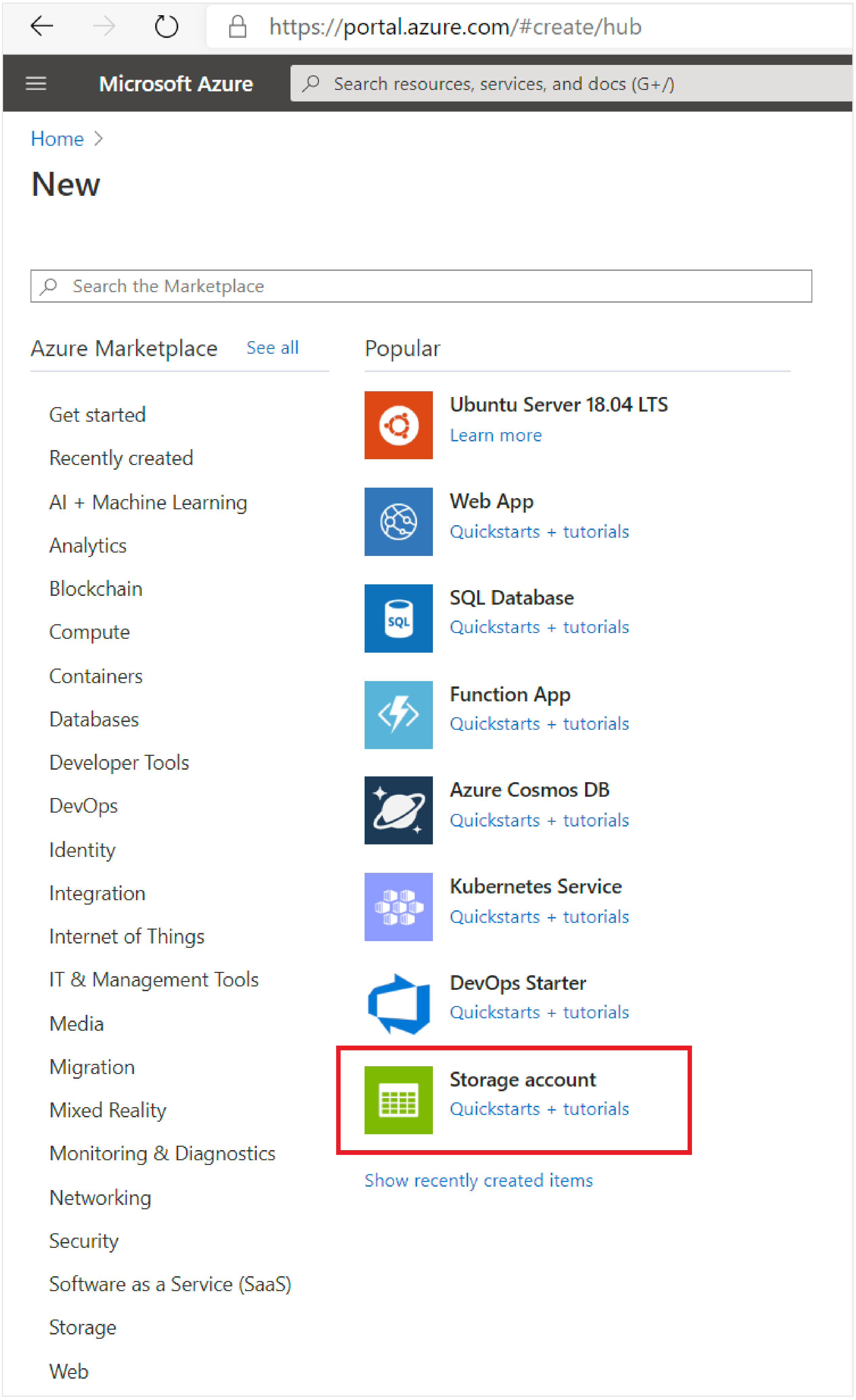
The simplest way to create a table in Azure Table Storage is to use the Azure portal. Follow these steps:

1. Sign into the Azure portal using your Azure account.

2. On the home page of the Azure portal, select **+Create a resource**.



3. On the **New** page, select **Storage account - blob, file, table, queue**



https://portal.azure.com/#create/hub

Microsoft Azure

Search resources, services, and docs (G+/)

Home >

New

Search the Marketplace

Azure Marketplace See all

- Get started
- Recently created
- AI + Machine Learning
- Analytics
- Blockchain
- Compute
- Containers
- Databases
- Developer Tools
- DevOps
- Identity
- Integration
- Internet of Things
- IT & Management Tools
- Media
- Migration
- Mixed Reality
- Monitoring & Diagnostics
- Networking
- Security
- Software as a Service (SaaS)
- Storage
- Web

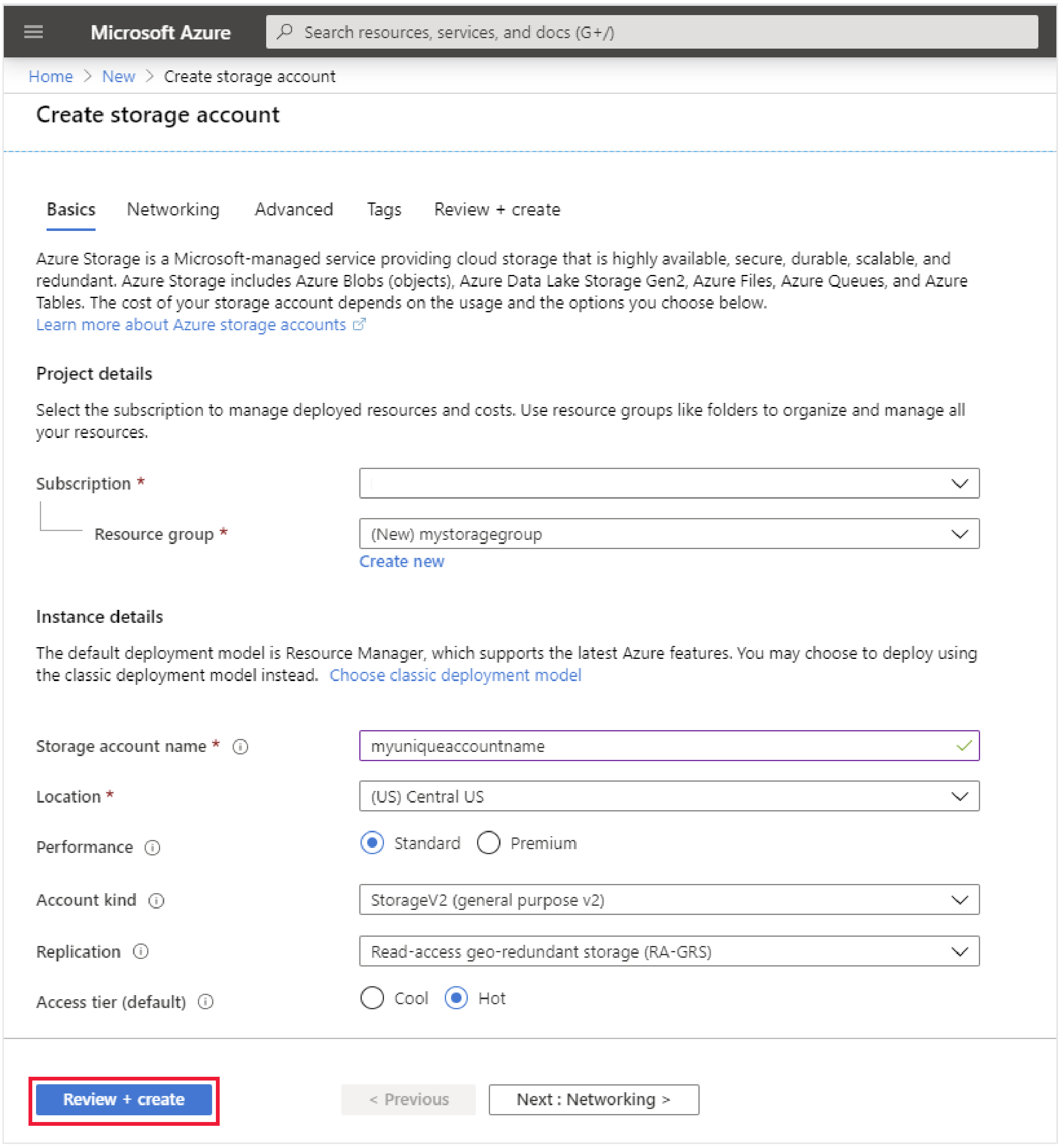
Popular

- Ubuntu Server 18.04 LTS — Learn more
- Web App — Quickstarts + tutorials
- SQL Database — Quickstarts + tutorials
- Function App — Quickstarts + tutorials
- Azure Cosmos DB — Quickstarts + tutorials
- Kubernetes Service — Quickstarts + tutorials
- DevOps Starter — Quickstarts + tutorials
- Storage account — Quickstarts + tutorials

Show recently created items

4. On the **Create storage account** page, enter the following details, and then select **Review + create**.

Field	Value
Subscription	Select your Azure subscription
Resource group	Select **Create new**, and specify the name of a new Azure resource group. Use a name of your choice, such as *mystoragegroup*
Storage account name	Enter a name of your choice for the storage account. The name must be unique though
Location	Select your nearest location
Performance	Standard
Account kind	StorageV2 (general purpose v2)
Replication	Read-access geo-redundant storage (RA-GRS)
Access tier	Hot


Microsoft Azure
Search resources, services, and docs (G+/)
Home > New > Create storage account

Create storage account

Basics Networking Advanced Tags Review + create

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

Project details

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

Subscription *	
Resource group *	(New) mystoragegroup

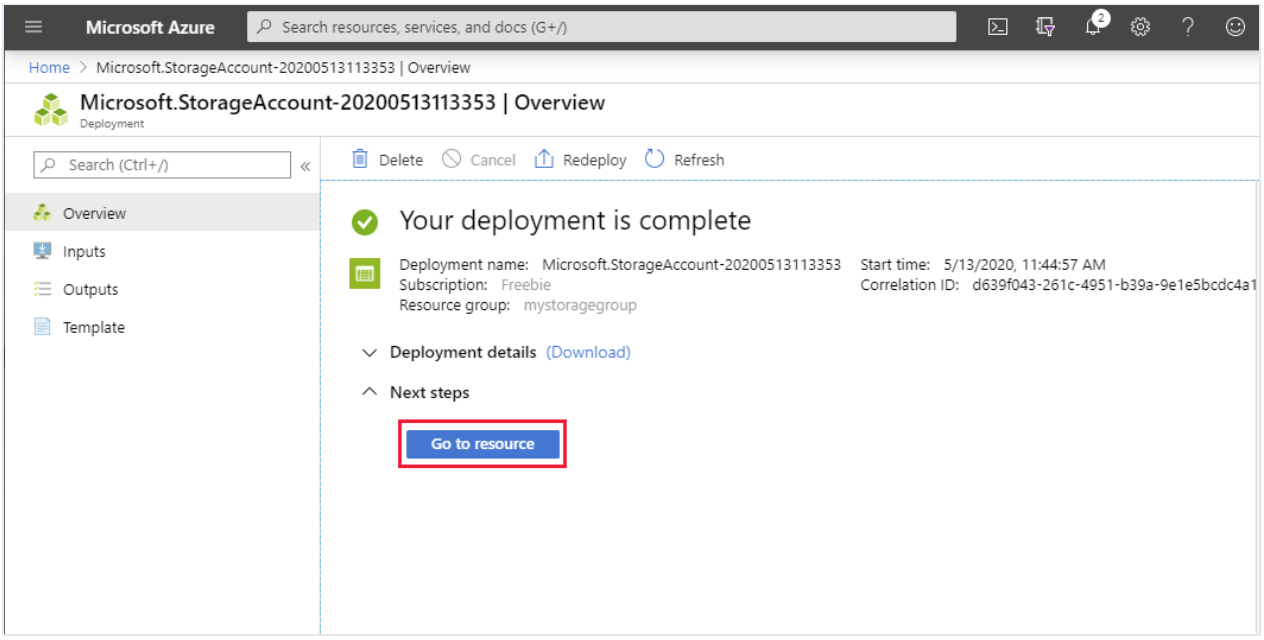
Instance details

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 *	myuniqueaccountname
Location *	(US) Central US
Performance	[✓ Standard] Premium
Account kind	StorageV2 (general purpose v2)
Replication	Read-access geo-redundant storage (RA-GRS)
Access tier (default)	Cool [✓ Hot]

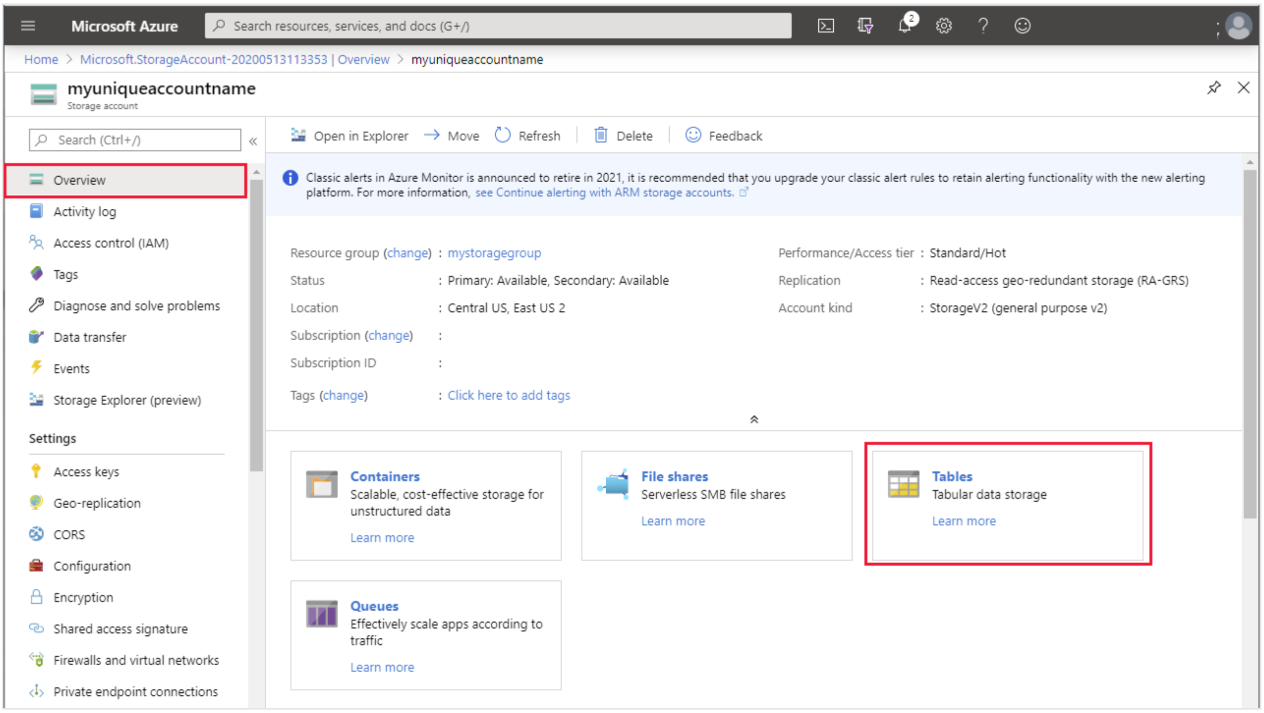
5. On the validation page, click **Create**, and wait while the new storage account is configured.

6. When the **Your deployment is complete** page appears, select **Go to resource**.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.StorageAccount-20200513113353 | Overview
Microsoft.StorageAccount-20200513113353 | Overview
Deployment
Search (Ctrl+/)
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Refresh
Your deployment is complete
Deployment details (Download)
Next steps
Go to resource

Deployment name: Microsoft.StorageAccount-20200513113353
Subscription: Freebie
Resource group: mystoragegroup
Start time: 5/13/2020, 11:44:57 AM
Correlation ID: d639f043-261c-4951-b39a-9e1e5bcdc4a1

7. On the **Overview** page for the new storage account, select **Tables**.

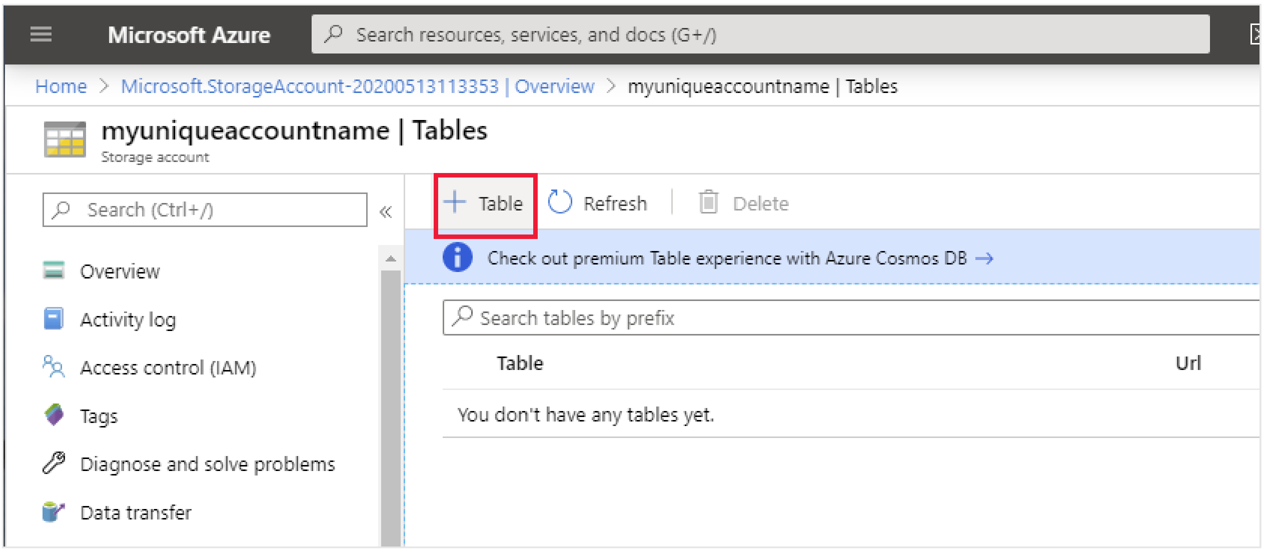


Classic alerts in Azure Monitor is announced to retire in 2021, it is recommended that you upgrade your classic alert rules to retain alerting functionality with the new alerting platform. For more information, see Continue alerting with ARM storage accounts.

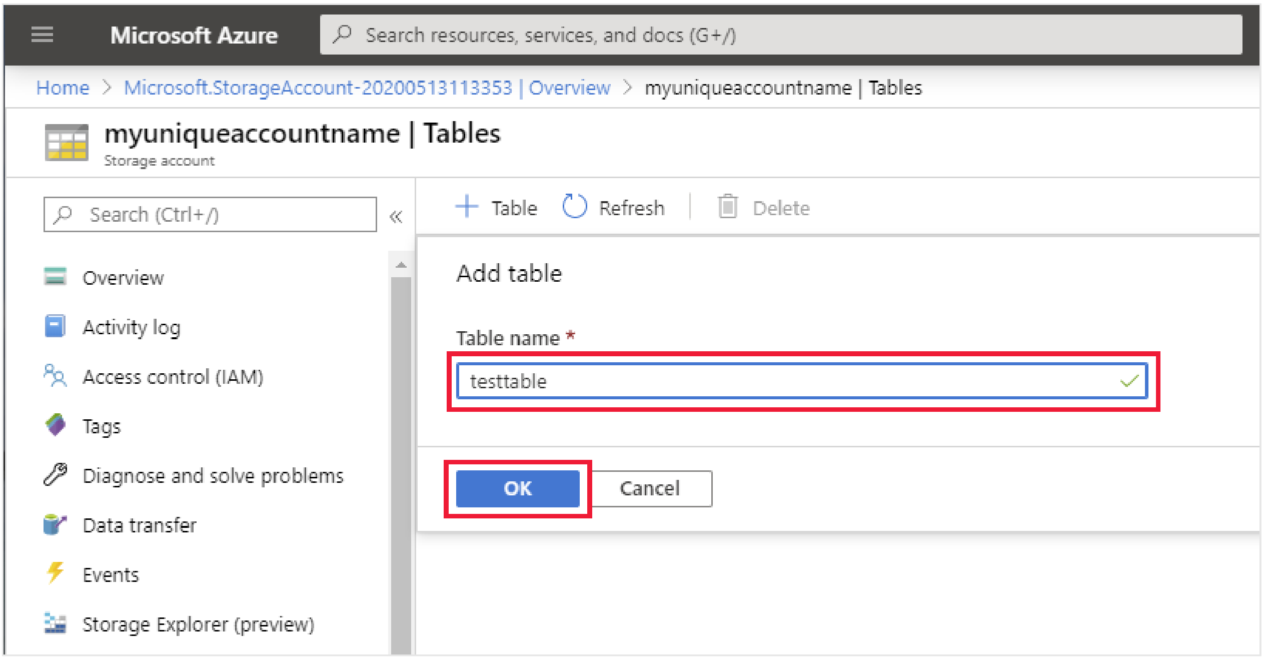
Resource group (change) : mystoragegroup
Status : Primary: Available, Secondary: Available
Location : Central US, East US 2
Subscription (change) :
Subscription ID :
Tags (change) : Click here to add tags
Performance/Access tier : Standard/Hot
Replication : Read-access geo-redundant storage (RA-GRS)
Account kind : StorageV2 (general purpose v2)

Containers — Scalable, cost-effective storage for unstructured data — Learn more
File shares — Serverless SMB file shares — Learn more
Tables — Tabular data storage — Learn more
Queues — Effectively scale apps according to traffic — Learn more

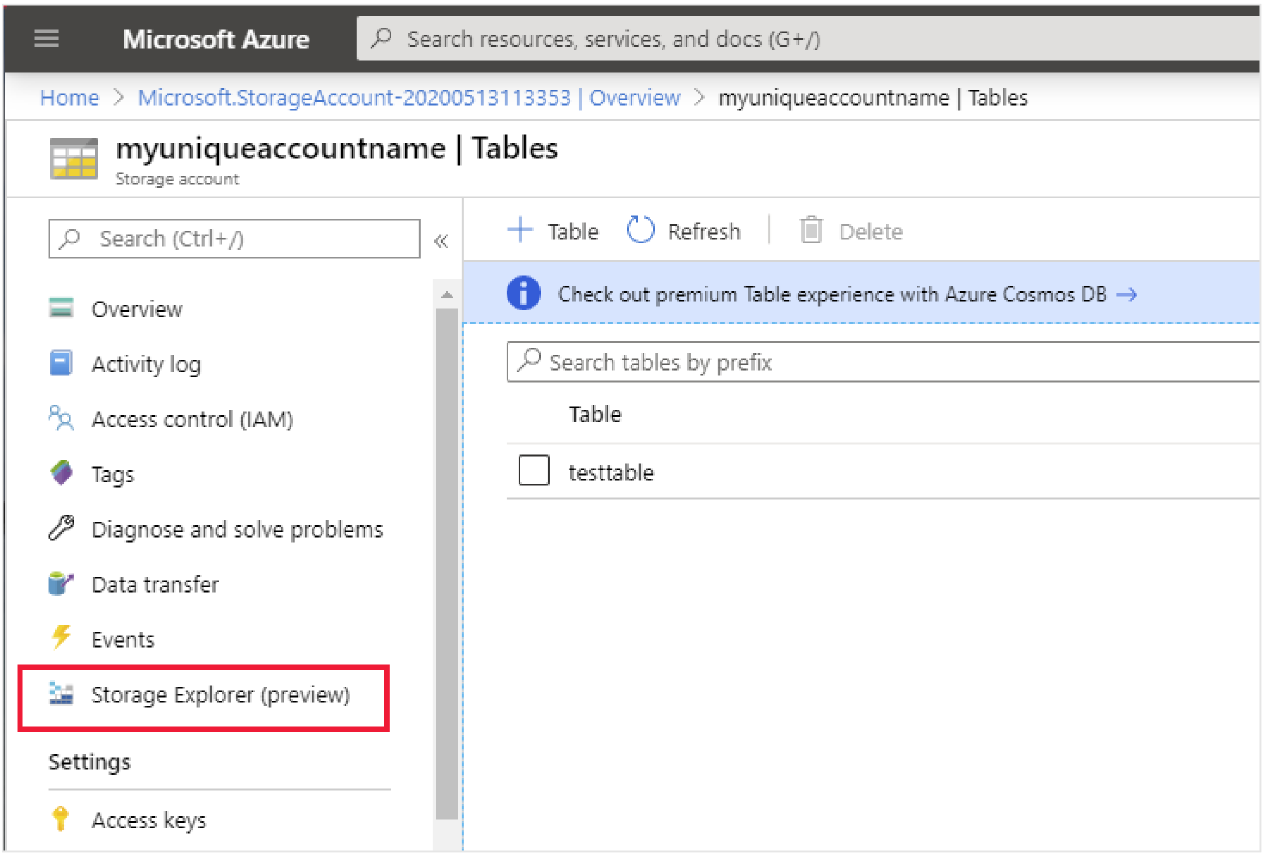
8. On the **Tables** page, select **+ Table**.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.StorageAccount-20200513113353 | Overview > myuniqueaccountname | Tables
myuniqueaccountname | Tables
Storage account
Search (Ctrl+/)
+ Table
Refresh
Delete
Check out premium Table experience with Azure Cosmos DB →
Search tables by prefix
Table
Url
You don't have any tables yet.
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Data transfer

9. In the **Add table** dialog box, enter **testtable** for the name of the table, and then select **OK**.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.StorageAccount-20200513113353 | Overview > myuniqueaccountname | Tables
myuniqueaccountname | Tables
Storage account
Search (Ctrl+/)
+ Table
Refresh
Delete
Add table
Table name *
testtable
OK
Cancel
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Data transfer
Events
Storage Explorer (preview)

10. When the new table has been created, select **Storage Explorer**.

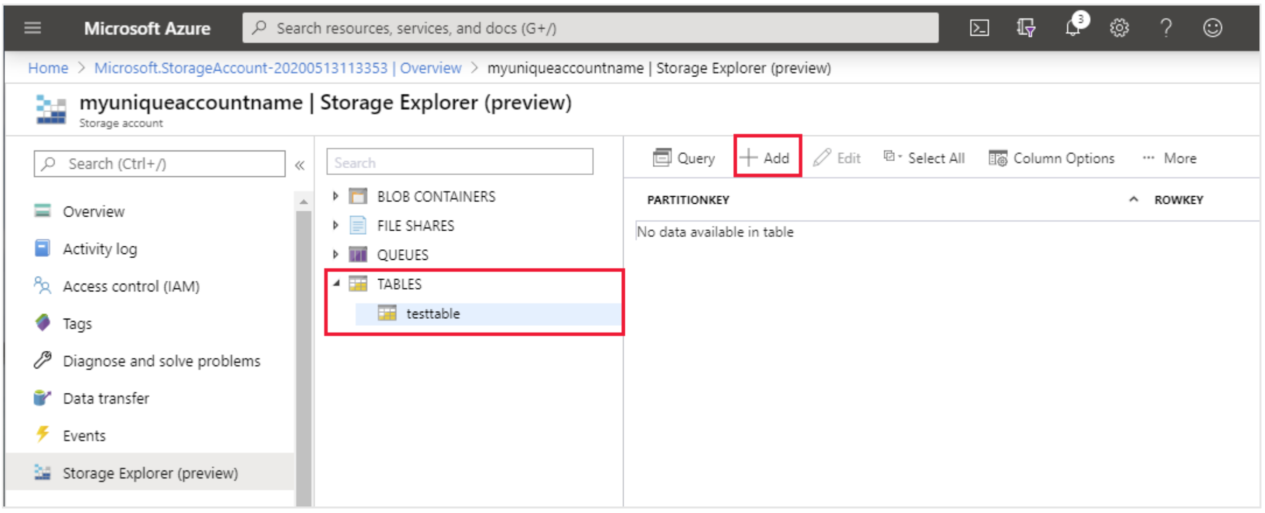

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.StorageAccount-20200513113353 | Overview > myuniqueaccountname | Tables
myuniqueaccountname | Tables
Storage account
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Data transfer
Events
Storage Explorer (preview)
Settings
Access keys
Table
Refresh
Delete
Search tables by prefix

Check out premium Table experience with Azure Cosmos DB →

Table
testtable

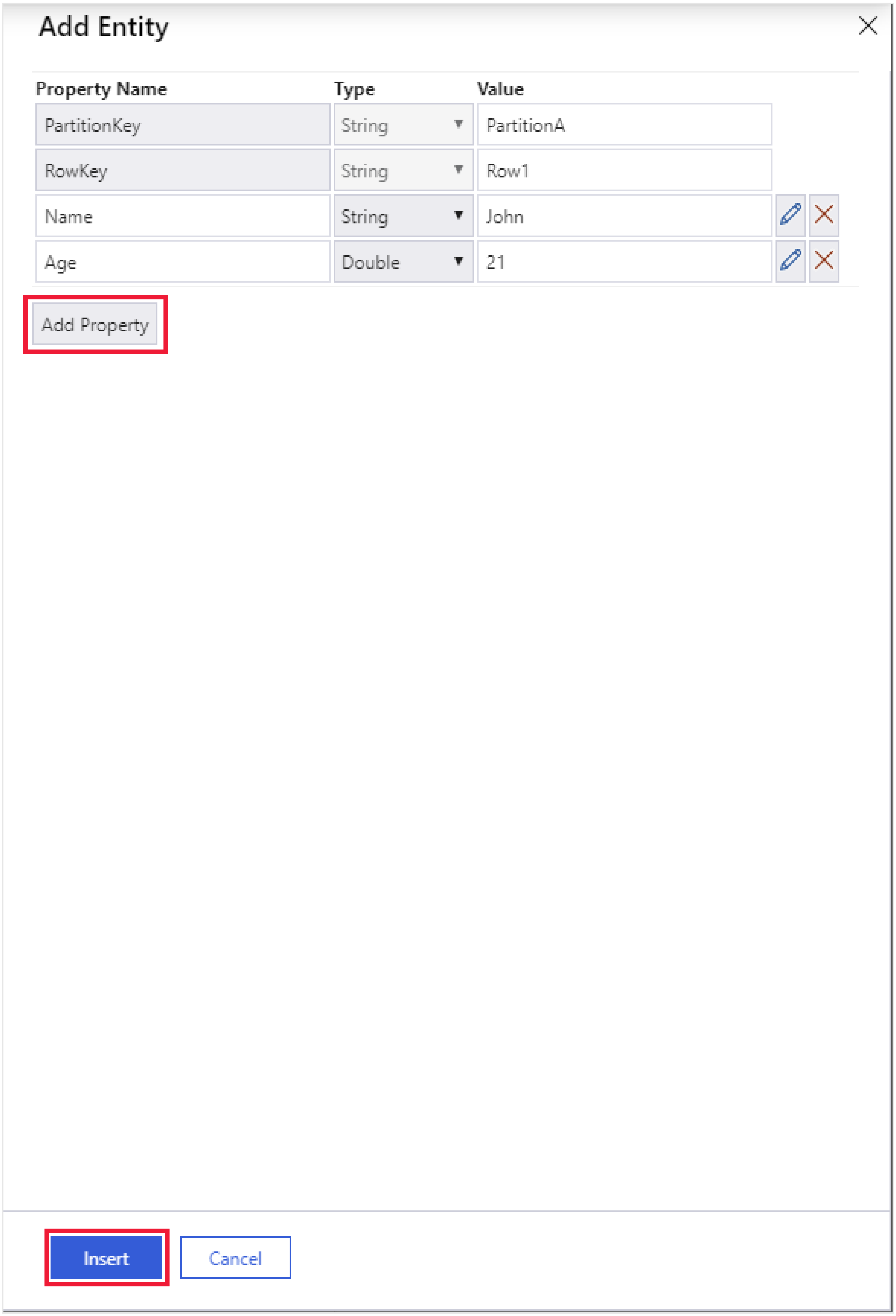
11. On the **Storage Explorer** page, expand **Tables**, and then select **testtable**. Select **Add** to insert a new entity into the table.

> **Note**
>
> In **Storage Explorer**, rows are also called *entities*.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.StorageAccount-20200513113353 | Overview > myuniqueaccountname | Storage Explorer (preview)
myuniqueaccountname | Storage Explorer (preview)
Storage account
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Data transfer
Events
Storage Explorer (preview)
Search
BLOB CONTAINERS
FILE SHARES
QUEUES
TABLES
testtable
Query
Add
Edit
Select All
Column Options
More
PARTITIONKEY
ROWKEY

No data available in table

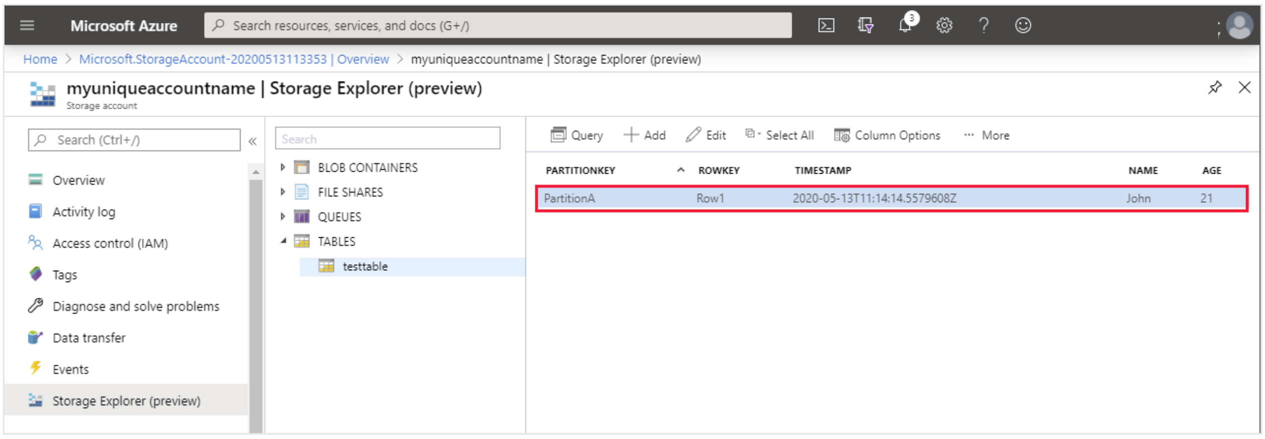
12. In the **Add Entity** dialog box, enter your own values for the **PartitionKey** and **RowKey** properties, and then select **Add Property**. Add a String property called **Name** and set the value to your name. Select **Add Property** again, and add a Double property (this is numeric) named **Age**, and set the value to your age. Select **Insert** to save the entity.



Add Entity

Property Name	Type	Value
PartitionKey	String	PartitionA
RowKey	String	Row1
Name	String	John
Age	Double	21

13. Verify that the new entity has been created. The entity should contain the values you specified, together with a timestamp that contains the date and time that the entity was created.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.StorageAccount-20200513113353 | Overview > myuniqueaccountname | Storage Explorer (preview)
myuniqueaccountname | Storage Explorer (preview)
Storage account
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Data transfer
Events
Storage Explorer (preview)
Search
BLOB CONTAINERS
FILE SHARES
QUEUES
TABLES
testtable
Query
Add
Edit
Select All
Column Options
More

PARTITIONKEY	ROWKEY	TIMESTAMP	NAME	AGE
PartitionA	Row1	2020-05-13T11:14:14.5579608Z	John	21

14. If time allows, experiment with creating additional entities. Not all entities must have the same properties. You can use the **Edit** function to modify the values in entity, and add or remove properties. The **Query** function enables you to find entities that have properties with a specified set of values.

Next unit: Explore Azure Blob storage

Continue >