😊 Azure Queue Storage

Azure Queue Storage is a service provided by Microsoft Azure that enables the creation and use of message queues to store large numbers of messages that can be accessed from anywhere via authenticated calls. It is designed for scenarios where you need to decouple components of an application and allow them to communicate asynchronously.

Key Features of Azure Queue Storage:

1. **Simple Messaging**: Azure Queue Storage is used for simple, temporary message storage between application components.
2. **Large Volumes**: It can handle large volumes of messages, with each message up to 64 KB. The total capacity of a queue is up to 200 TB.
3. **Asynchronous Communication**: It supports asynchronous messaging, which helps decouple a distributed application's components.
4. **Access Anywhere**: Messages can be accessed from anywhere via authenticated calls using HTTP or HTTPS.
5. **REST API Support**: The service provides a REST-based interface, enabling easy integration with various applications and services.
6. **Time-to-Live (TTL)**: Messages in the queue can have a TTL value, after which they will be automatically deleted.
7. **Poison Message Handling**: Support for detecting and handling messages that repeatedly fail processing.

Common Use Cases:

1. **Decoupling Application Components**: Useful in microservices architecture where different services need to communicate without being tightly coupled.
2. **Load Levelling**: Smooth out intermittent heavy traffic by using queues to buffer the load and process it at a steady rate.
3. **Task Scheduling**: Schedule tasks to be processed asynchronously, such as background job processing.
4. **Event Queueing**: Use queues to manage events and ensure they are processed in the correct order and without duplication.

How It Works:

1. **Creating a Queue**: A queue is created within a storage account.
2. **Adding Messages**: Messages are added to the queue using the PutMessage operation.
3. **Reading Messages**: Messages are read from the queue using the GetMessage operation. This makes the message invisible to other readers for a specified period.

4. **Processing and Deleting Messages**: Once processed, the message is deleted from the queue using the DeleteMessage operation.

Benefits:

1. **Scalability**: Can handle large amounts of messages and scale as needed.
2. **Reliability**: Ensures messages are delivered reliably and processed in the correct order.
3. **Cost-Effective**: Offers a cost-effective way to implement asynchronous messaging in applications.
4. **Ease of Integration**: Easily integrates with other Azure services and on-premises applications.

Azure Queue Storage is an essential component for building scalable, reliable, and decoupled cloud applications.

🤔 What are we doing in this lab?

In this process, we are setting up and utilizing Azure Queue Storage to manage and process messages in an organized, asynchronous manner. Here's a breakdown of the steps and the end goal:

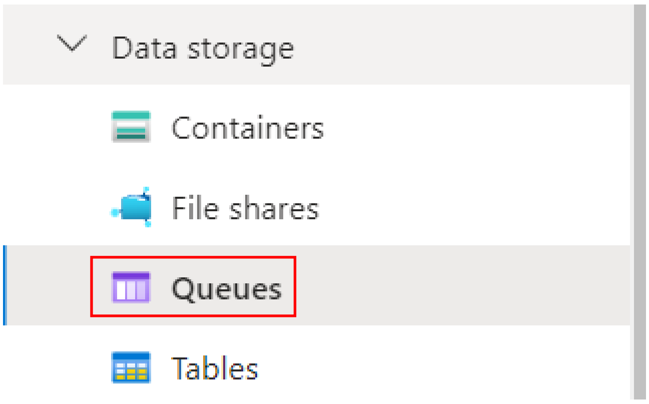
1. **Azure Portal Login**: We log in to the Azure Portal to manage our cloud resources.
2. **Creating a Storage Account**: We create a storage account, which serves as a container for our queue and other storage services.
3. **Navigating to Queue Storage**: Within the storage account, we navigate to the Queue section under data storage.
4. **Creating a Queue**: We create a new queue named "orders" where our messages (representing orders) will be stored.
5. **Adding Messages to the Queue**: We add messages to the queue, each representing an order (e.g., shirts) with an expiration of 7 days.
6. **FIFO Processing**: The queue operates on a First-In-First-Out (FIFO) basis, ensuring that the first message added is the first to be processed.

End Goal

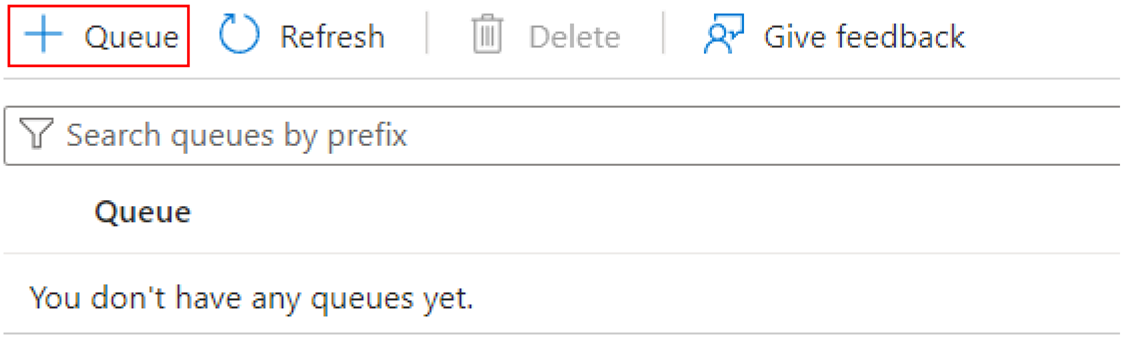
The ultimate goal is to set up a robust messaging system that allows different components of an application to communicate asynchronously. This ensures efficient handling of tasks, load leveling, and decoupling of services, which is crucial for building scalable and reliable applications. By following these steps, we create a queue storage system that can manage orders efficiently, enabling smooth and orderly processing in a real-world application environment.

😄 To begin with this Lab

1. Now to work with Azure queue storage first we need to go to Azure Portal and login, then we need to create a storage account.
2. Once we have the storage account then we need to go into it. Then from the left pane expand data storage and go to Queue.

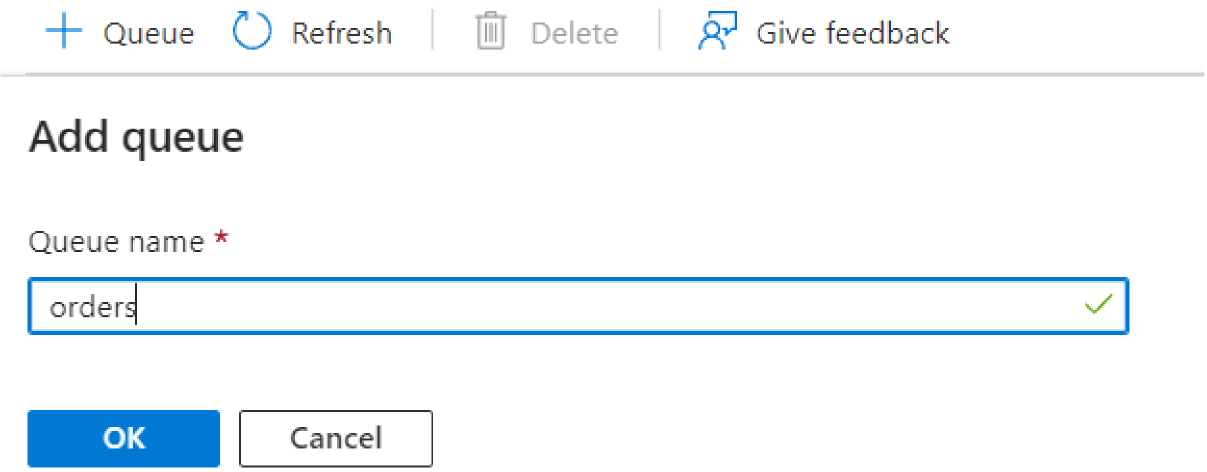


3. Now to create a queue you need to click on add queue which is highlighted below.



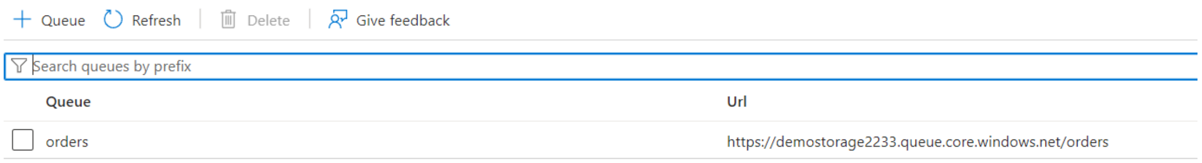
You don't have any queues yet.

4. Then just give it a name and click on OK.



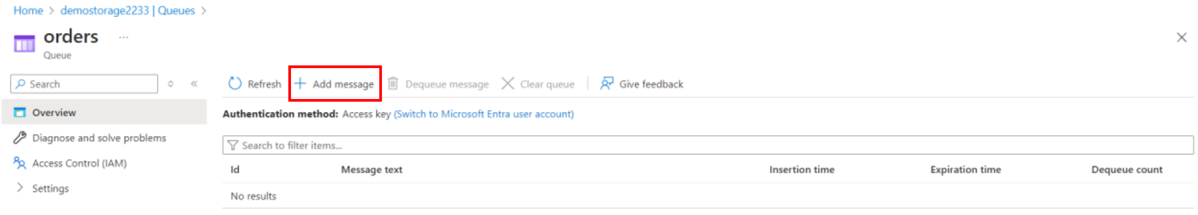
Queue name * orders

5. Below you can see that our queue has been created. Now to work further we need to click on our queue i.e. orders and go inside of it.



Queue	Url
□ orders	https://demostorage2233.queue.core.windows.net/orders

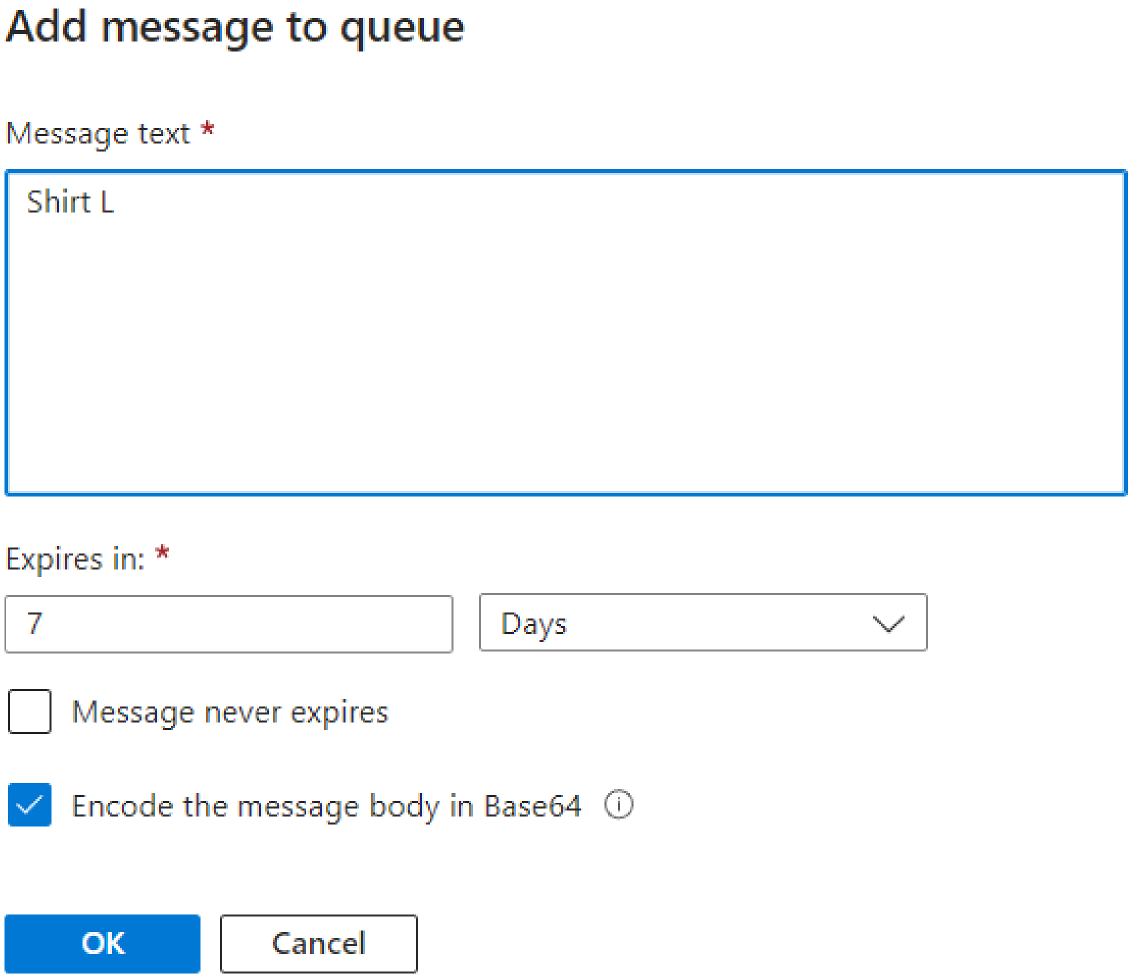
6. Now you can see that we do not have any data in it. So, now we will have some messages in it. For that click on add message.



Authentication method: Access key (Switch to Microsoft Entra user account)

Id	Message text	Insertion time	Expiration time	Dequeue count
No results				

7. So, below you can see that we can add message in text format. Now our queue is order type which is why we have added an order of shirts, also this message will expire in 7 days automatically. Then just click on OK.



Add message to queue

Message text *

Shirt L

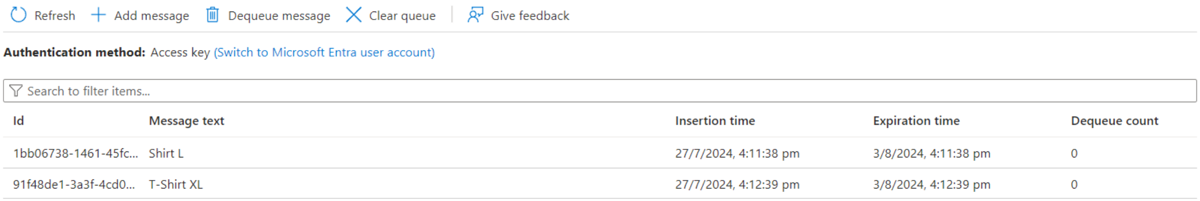
Expires in: *

7 Days

□ Message never expires

☑ Encode the message body in Base64

8. Likewise, we added one add one more message in it which you can see below.



Authentication method: Access key (Switch to Microsoft Entra user account)

Id	Message text	Insertion time	Expiration time	Dequeue count
1bb06738-1461-45fc...	Shirt L	27/7/2024, 4:11:38 pm	3/8/2024, 4:11:38 pm	0
91f48de1-3a3f-4cd0...	T-Shirt XL	27/7/2024, 4:12:39 pm	3/8/2024, 4:12:39 pm	0

9. Also, this works on first in first out (FIFO). So, this is a basic overview of how Queue as a storage works in Azure.
10. More importantly in real life we do not add messages directly into queue it is done by a third-party application who uses queue storage as a service.