**Queue & Tables**

* **Queue**

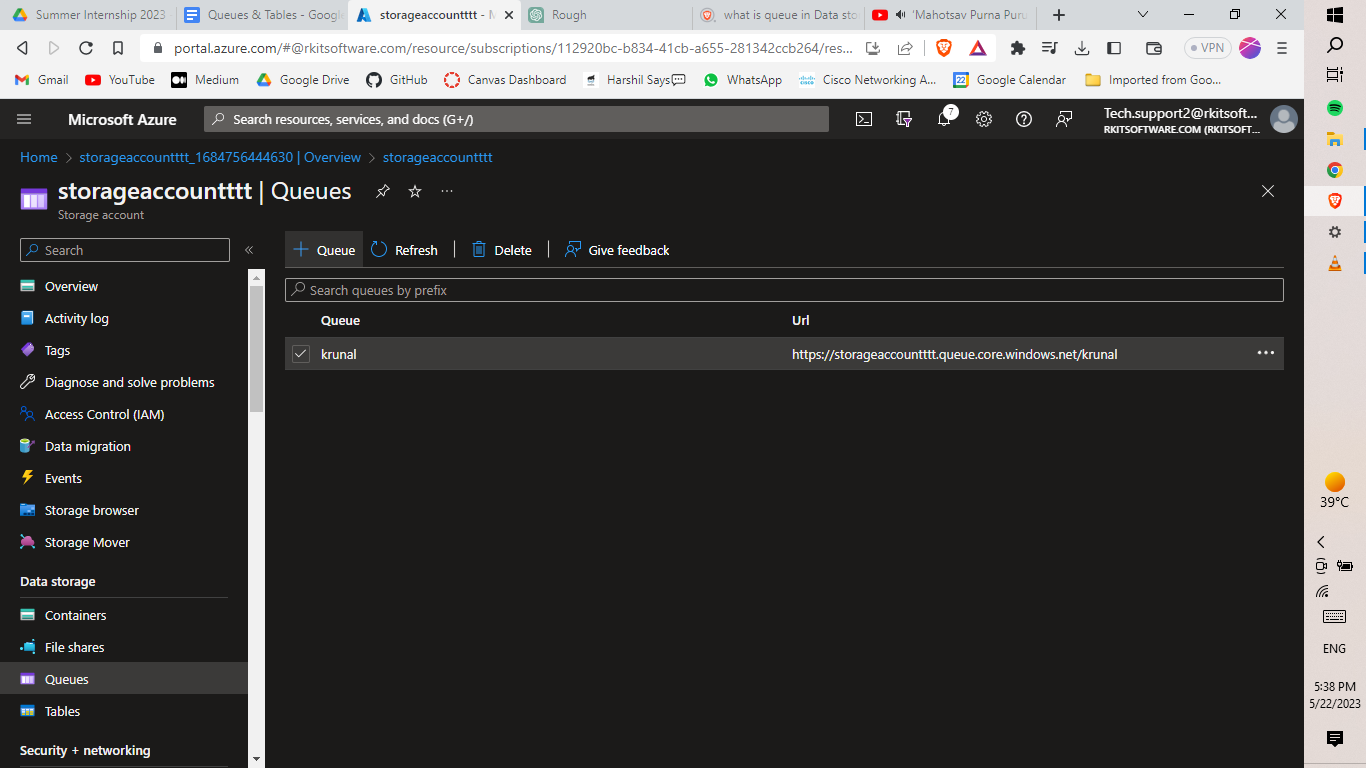
In Azure, a queue is a storage service provided by Azure Storage for storing and retrieving messages. It is a part of Azure's data storage offerings and is designed for building scalable and reliable applications.

A queue allows you to decouple components of your application by providing a reliable and asynchronous communication mechanism. Messages in the queue can be accessed in a first-in, first-out (FIFO) order, meaning the messages are processed in the same order they were added to the queue.

Common scenarios where Azure Queue Storage is used include decoupling workloads, task distribution, message passing between components, and building fault-tolerant systems.

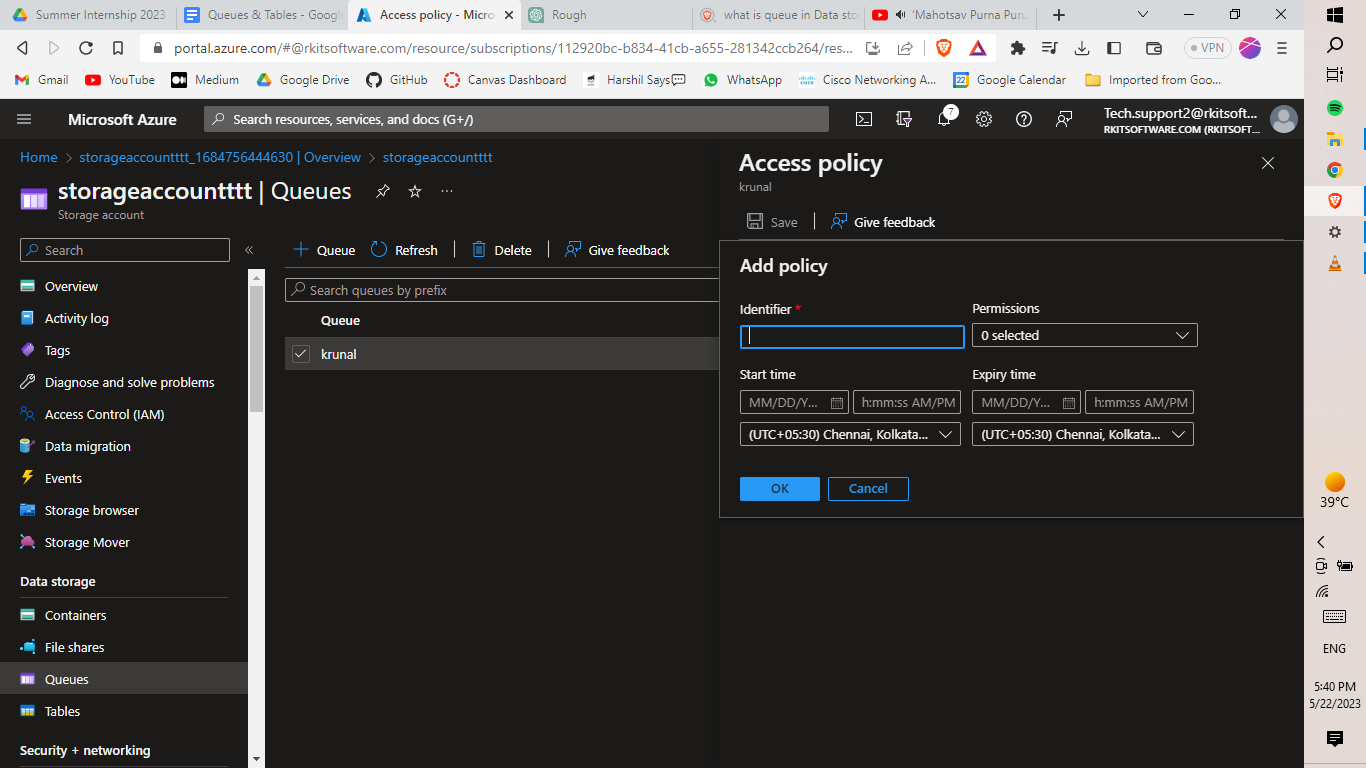
To use Azure Queue Storage, you need to create a storage account in Azure and then create queues within that storage account. You can interact with the queues using Azure SDKs, REST APIs, or client libraries for various programming languages.

Overall, Azure Queue Storage provides a reliable and scalable messaging solution for building distributed applications and enables loose coupling and asynchronous communication between components.

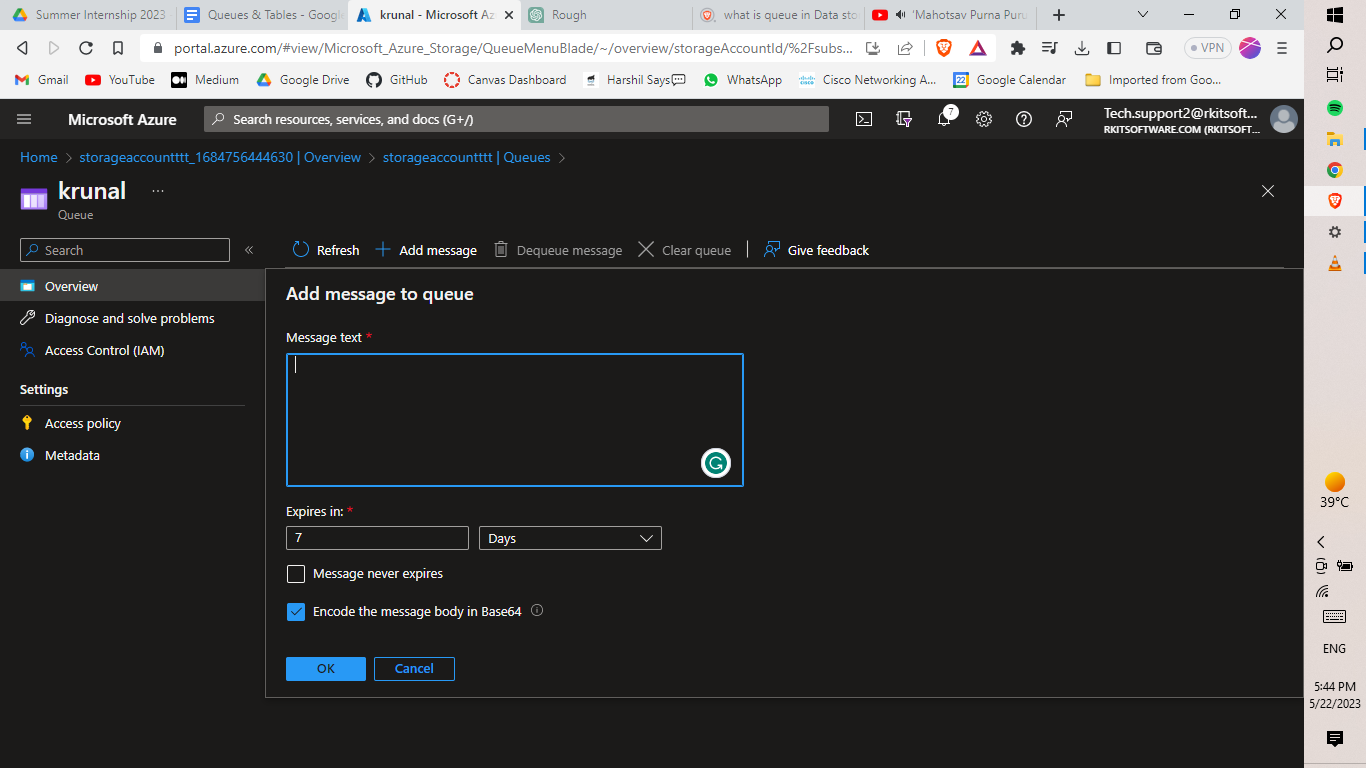


Here, by clicking on the +Queue button we will be able to create a queue in our storage account.

Azure Storage Queue is a service hosted on the Azure platform used for storing large numbers of messages.



And when you click on the three dots right side, you will be able to define the policy and who has the right to read - add - update & process of that content. Along with that, we can also define the start time and end time of that policy.



When you click on that Queue you will be able to add the message to the queue.

Azure Storage Queues are designed for very large cloud networks or hybrid networks, providing a highly reliable and inexpensive queuing service. A single message can be up to 64 KB in size, and a queue can hold millions of messages, up to the total capacity limit of a storage account (200 TB).

**Use cases:**

* Asynchronous Processing: Queues are commonly used for asynchronous processing, where messages can be added to the queue by one component of the application and processed by another component at a later time.
* Reliability and Durability: Azure Queue Storage ensures the high availability and durability of messages.
* Message Visibility: When a message is retrieved from the queue, it becomes invisible to other processes for a configurable visibility timeout period.
* Scalability: Azure Queue Storage is designed to handle large volumes of messages with high throughput. It scales automatically to accommodate the workload and provides seamless horizontal scaling as your application grows.
* **Tables:**

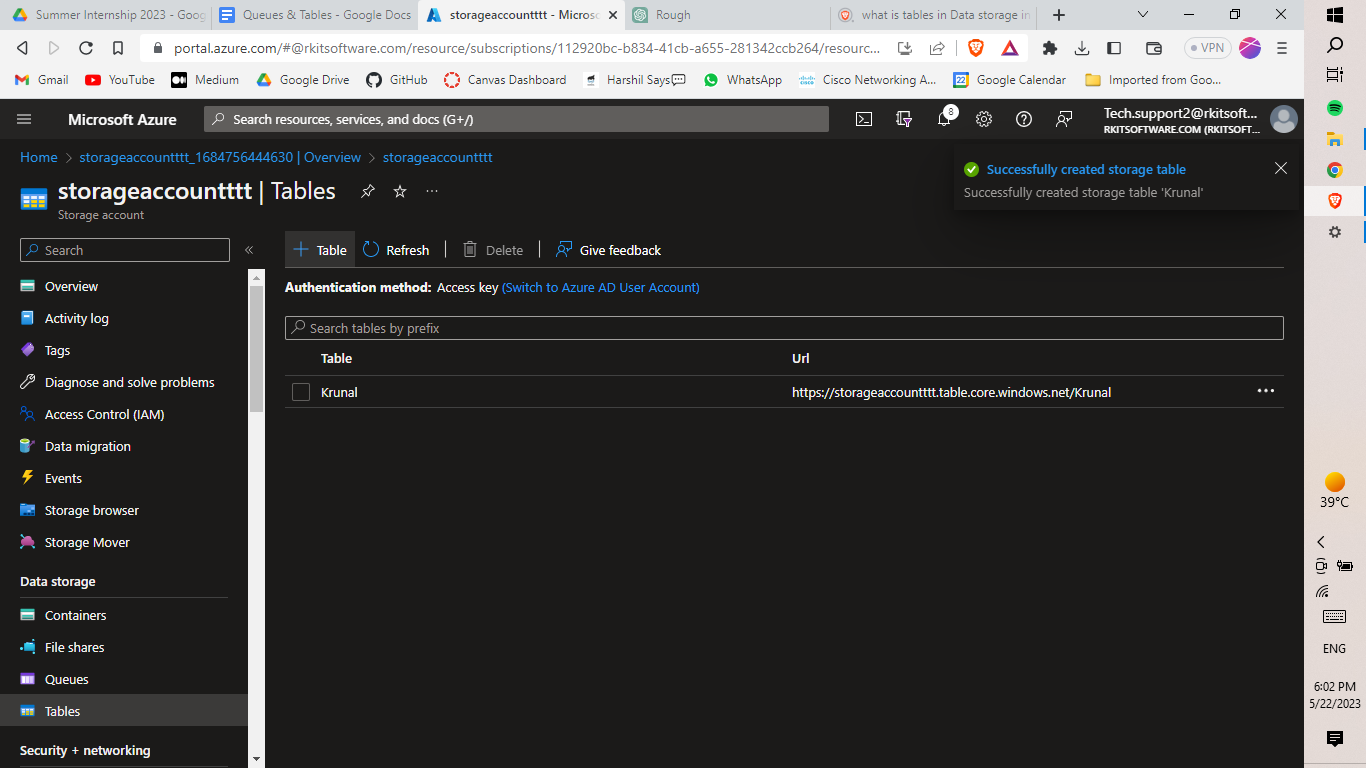
In Azure, tables are a type of NoSQL storage provided by Azure Table Storage. Azure Table Storage is a key-value store that allows you to store large amounts of structured data in a flexible schema-less format.

Tables in Azure Table Storage consist of a collection of entities, where each entity represents a row in the table. An entity is made up of properties, which are name-value pairs that define the data stored in the entity. Tables are designed for high scalability, allowing you to store and retrieve large amounts of data with high performance.

Azure Table Storage is commonly used for scenarios such as storing user profiles, logging data, sensor data, and other types of structured data. It is often used in combination with other Azure services to build scalable and reliable applications.

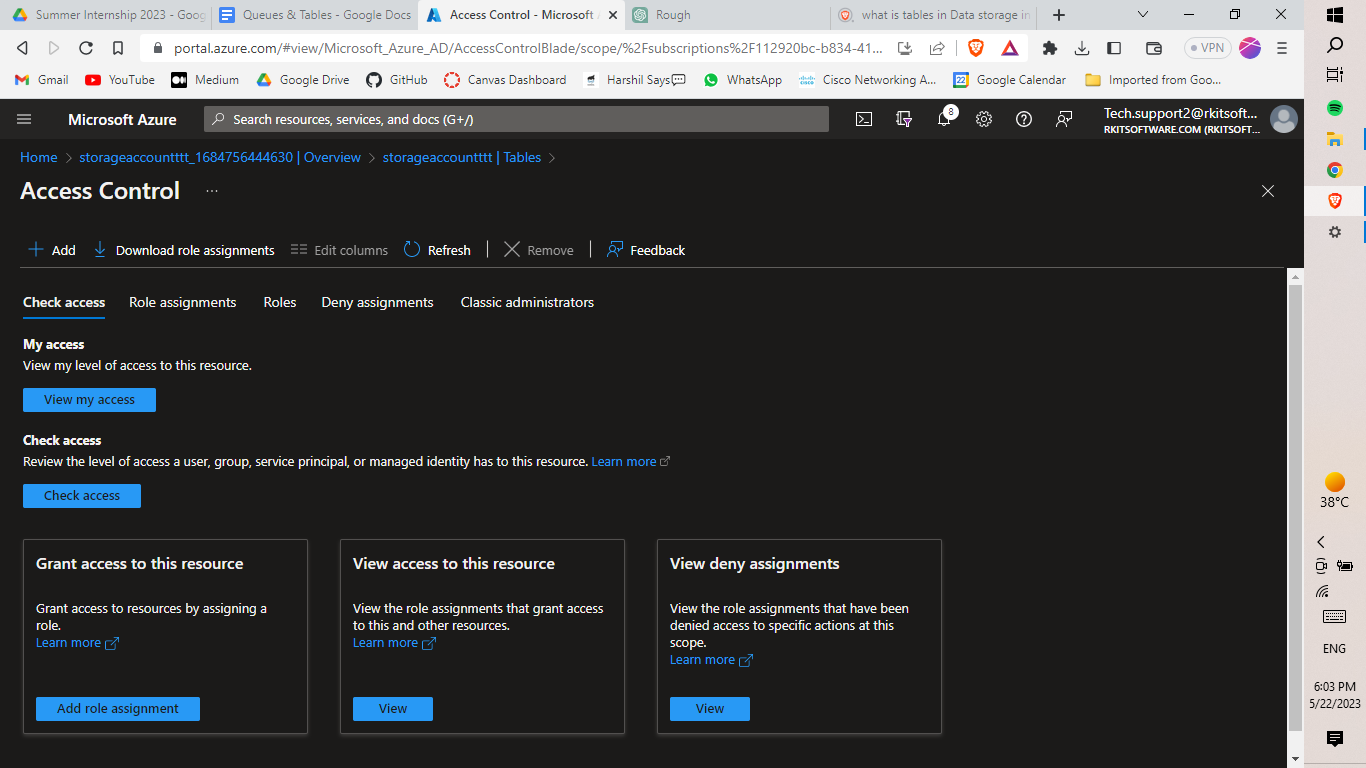
Azure Table Storage is a cloud-based NoSQL-like database that can be used for storage and data operations on structured or semi-structured data.

It offers a schematic design, allowing for the storage of a collection of entities in one table, each containing a set of properties that define a name-value pair.



Here, by clicking on the +Table button we will be able to create a table in our storage account.

And when you click on the three dots it will show the Identity and Access Management (IAM) of that particular table.



By configuring all the steps, you will be able to define the particular roles of the respective users.

**Use cases:**

* Schema-less: Tables in Azure Table Storage are schema-less, meaning that different entities in the same table can have different sets of properties. This flexibility makes it suitable for storing diverse data sets.
* Scalability: Azure Table Storage is designed to handle large amounts of data and high read/write throughput. It can scale to accommodate massive workloads and provides automatic load balancing.
* Querying and Indexing: Azure Table Storage supports querying data using the OData protocol, allowing you to perform various filtering and projection operations on the data.
* Durability and Availability: Azure Table Storage provides durability and availability for your data by automatically replicating it across multiple data centers. This ensures data resilience and high availability.
* Cost-effective: Azure Table Storage is cost-effective compared to other storage options in Azure.