# Performance levels in DocumentDB

This article provides an overview of performance levels in [Microsoft Azure DocumentDB](/services/documentdb/).

After reading this article, you’ll be able to answer the following questions:

* What is a performance level?
* How is throughput reserved for a database account?
* How do I work with performance levels?
* How am I billed for performance levels?

## Introduction to performance levels

Each DocumentDB collection created under a Standard account is provisioned with an associated performance level. Each collection in a database can have a different performance level allowing you to designate more throughput for frequently accessed collections and less throughput for infrequently accessed collections. DocumentDB supports both user-defined performance levels and pre-defined performance levels.

Each performance level has an associated [request unit (RU)](http://go.microsoft.com/fwlink/?LinkId=735027) rate limit. This is the throughput that will be reserved for a collection based on its performance level, and is available for use by that collection exclusively.

<tbody>  
 <tr>  
 <td valign="top"><p></p></td>  
 <td valign="top"><p>Details</p></td>  
 <td valign="top"><p>Throughput Limits</p></td>  
 <td valign="top"><p>Storage Limits</p></td>  
 <td valign="top"><p>Version</p></td>  
 <td valign="top"><p>APIs</p></td>   
 </tr>  
 <tr>  
 <td valign="top"><p>User-defined performance</p></td>  
 <td valign="top"><p>Storage metered based on usage in GB.</p><p>Throughput in units of 100 RU/s</p></td>  
 <td valign="top"><p>Unlimited. 400 - 250,000 request units/s by default (higher by request)</p></td>  
 <td valign="top"><p>Unlimited. 250 GB by default (higher by request) </p></td>  
 <td valign="top"><p>V2</p></td>  
 <td valign="top"><p>API 2015-12-16 and newer</p></td>   
 </tr>  
 <tr>  
 <td valign="top"><p>Pre-defined performance</p></td>  
 <td valign="top"><p>10 GB reserved storage.</p><p>S1 = 250 RU/s, S2 = 1000 RU/s, S3 = 2500 RU/s</p></td>  
 <td valign="top"><p>2500 RU/s</p></td>  
 <td valign="top"><p>10 GB</p></td>  
 <td valign="top"><p>V1</p></td>  
 <td valign="top"><p>Any</p></td>   
 </tr>   
</tbody>

DocumentDB allows for a rich set of database operations including queries, queries with user-defined functions (UDFs), stored procedures and triggers. The processing cost associated with each of these operations will vary based on the CPU, IO and memory required to complete the operation. Instead of thinking about and managing hardware resources, you can think of a request unit as a single measure for the resources required to perform various database operations and service an application request.

Collections can be created through the [Microsoft Azure portal](https://portal.azure.com), the [REST API](https://msdn.microsoft.com/library/azure/mt489078.aspx) or any of the [DocumentDB SDKs](https://msdn.microsoft.com/library/azure/dn781482.aspx). The DocumentDB APIs allow you to specify the performance level of a collection.

[AZURE.NOTE] The performance level of a collection can be adjusted through the APIs or the [Microsoft Azure portal](https://portal.azure.com/). Performance level changes are expected to complete within 3 minutes.

## Setting performance levels for collections

Once a collection is created, the full allocation of RUs based on the designated performance level are reserved for the collection.

Note that with both user-defined and pre-defined performance levels, DocumentDB operates based on reservation of throughput. By creating a collection, an application has reserved and is billed for reserved throughput regardless of how much of that throughput is actively used. With user-defined performance levels, storage is metered based on consumption, but with pre-defined performance levels, 10 GB of storage is reserved at the time of collection creation.

After collections are created, you can modify the performance level through the DocumentDB SDKs or through the Azure Classic Portal.

[AZURE.IMPORTANT] DocumentDB Standard collections are billed at an hourly rate and each collection you create will be billed for a minimum one hour of usage.

If you adjust the performance level of a collection within an hour, you will be billed for the highest performance level set during the hour. For example, if you increase your performance level for a collection at 8:53am you will be charged for the new level starting at 8:00am. Likewise, if you decrease your performance level at 8:53am, the new rate will be applied at 9:00am.

Request units are reserved for each collection based on the performance level set. Request unit consumption is evaluated as a per second rate. Applications that exceed the provisioned request unit rate (or performance level) on a collection will be throttled until the rate drops below the reserved level for that collection. If your application requires a higher level of throughput, you can increase the performance level for each collection.

[AZURE.NOTE] When your application exceeds performance levels for one or multiple collections, requests will be throttled on a per collection basis. This means that some application requests may succeed while others may be throttled. It is recommended to add

## Working with performance levels

DocumentDB collections allow you to group your data based on both the query patterns and performance needs of your application. With DocumentDB’s automatic indexing and query support, it is quite common to collocate heterogeneous documents within the same collection. The key considerations in deciding whether separate collections should be used include:

* Queries – A collection is the scope for query execution. If you need to query across a set of documents, the most efficient read patterns come from collocating documents in a single collection.
* Transactions – All transactions are scoped to within a single collection. If you have documents that must be updated within a single stored procedure or trigger, they must be stored within the same collection. More specifically, a partition key within a collection is the transaction boundary. Please see [Partitioning in DocumentDB](/documentation/articles/documentdb-partition-data) for more details.
* Performance isolation – A collection has an associated performance level. This ensures that each collection has a predictable performance through reserved RUs. Data can be allocated to different collections, with different performance levels, based on access frequency.

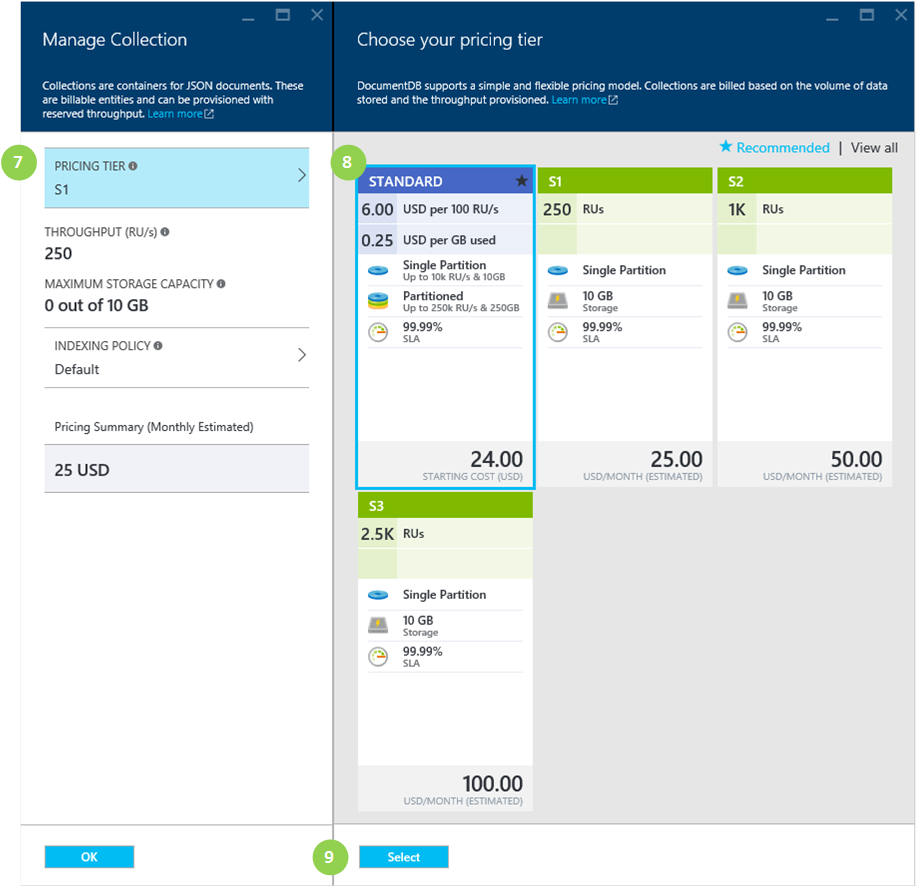
[AZURE.IMPORTANT] It is important to understand you will be billed at full standard rates based on the number of collections created by your application.

It is recommended that your application makes use of a small number of collections unless you have large storage or throughput requirements. Ensure that you have well understood application patterns for the creation of new collections. You may choose to reserve collection creation as a management action handled outside your application. Similarly, adjusting the performance level for a collection will change the hourly rate at which the collection is billed. You should monitor collection performance levels if your application adjusts these dynamically.

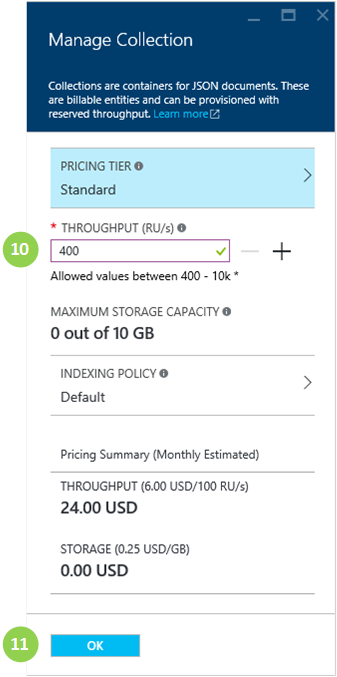
## Changing performance levels using the Azure Portal

The Azure Portal is one option available to you when managing your collections’ performance levels. Follow these steps to change from using pre-defined performance levels to user-defined performance levels in the Azure portal, or watch the 75 second [Channel 9 video](https://channel9.msdn.com/Blogs/AzureDocumentDB/ChangeDocumentDBCollectionPerformance). For more information about the change to the pricing options, see the blog post [DocumentDB: Everything you need to know about using the new pricing options](https://azure.microsoft.com/blog/documentdb-use-the-new-pricing-options-on-your-existing-collections/).

1. Navigate over to the [**Azure portal**](https://portal.azure.com) from your browser.
2. Click **Browse** from the jump bar on the left side.
3. In the **Browse** hub, click **DocumentDB Accounts** under the **Filter by** label.
4. In the **DocumentDB Accounts** blade, click the DocumentDB account that contains the desired collection.
5. In the **DocumentDB Account** blade, scroll down to the **Databases** lens and click the database that contains the desired collection.
6. In the newly opened **Database** blade, scroll down to the **Collections** lens and select your desired collection.
7. In the **Manage Collection** blade, click **Pricing tier**.

* 
* Screen shot of the Manage Collection and Choose your pricing tier blades for Azure DocumentDB showing where to change the pricing tier for the collection

1. In the **Choose your pricing tier** blade, click **Standard**.
2. In the **Choose your pricing tier** blade, click **Select**.
3. Back in the **Manage Collection** blade, the **Pricing Tier** was changed to **Standard** and the **Throughput (RU/s)** box is displayed.

* Change the value in the **Throughput** box to a value between 400 and 10,000 [Request units](documentdb-request-units.md)/second (RU/s). The **Pricing Summary** on the bottom of the page updates automatically to provide an estimate of the monthly cost.
* 
* Screen shot of the Manage Collection blade showing where to change the throughput value for the collection

1. On the **Manage Collection** blade, click **OK** to update your collection to the user-defined performance.

If you determine that you need more throughput (greater than 10,000 RU/s) or more storage (greater than 10GB) you can create a partitioned collection. To create a partitioned collection, see [Create a collection](/documentation/articles/documentdb-create-collection).

[AZURE.NOTE] Changing performance levels of a collection may take up to 2 minutes.

## Changing performance levels using the .NET SDK

Another option for changing your collections’ performance levels is through our SDKs. This section only covers changing a collection’s performance level using our [.NET SDK](https://msdn.microsoft.com/library/azure/dn948556.aspx), but the process is similar for our other [SDKs](https://msdn.microsoft.com/library/azure/dn781482.aspx). If you are new to our .NET SDK, please visit our [getting started tutorial](/documentation/articles/documentdb-get-started).

Here is a code snippet for changing the offer throughput to 50,000 request units per second:

//Fetch the resource to be updated  
Offer offer = client.CreateOfferQuery()  
 .Where(r => r.ResourceLink == collection.SelfLink)   
 .AsEnumerable()  
 .SingleOrDefault();  
   
// Set the throughput to 5000 request units per second  
offer = new OfferV2(offer, 5000);  
   
//Now persist these changes to the database by replacing the original resource  
await client.ReplaceOfferAsync(offer);  
  
// Set the throughput to S2  
offer = new Offer(offer);  
offer.OfferType = "S2";  
   
//Now persist these changes to the database by replacing the original resource  
await client.ReplaceOfferAsync(offer);

[AZURE.NOTE] Collections provisioned with under 10,000 request units per second can be migrated between offers with user-defined throughput and pre-defined throughput (S1, S2, S3) at any time. Collections which are provisioned with above 10,000 request units per second cannot be converted to pre-defined throughput levels.

Visit [MSDN](https://msdn.microsoft.com/library/azure/microsoft.azure.documents.client.documentclient.aspx) to view additional examples and learn more about our offer methods:

* [**ReadOfferAsync**](https://msdn.microsoft.com/library/azure/microsoft.azure.documents.client.documentclient.readofferasync.aspx)
* [**ReadOffersFeedAsync**](https://msdn.microsoft.com/library/azure/microsoft.azure.documents.client.documentclient.readoffersfeedasync.aspx)
* [**ReplaceOfferAsync**](https://msdn.microsoft.com/library/azure/microsoft.azure.documents.client.documentclient.replaceofferasync.aspx)
* [**CreateOfferQuery**](https://msdn.microsoft.com/library/azure/microsoft.azure.documents.linq.documentqueryable.createofferquery.aspx)

## Next steps

To learn more about pricing and managing data with Azure DocumentDB, explore these resources:

* [DocumentDB pricing](https://azure.microsoft.com/pricing/details/documentdb/)
* [Managing DocumentDB capacity](/documentation/articles/documentdb-manage)
* [Modeling data in DocumentDB](/documentation/articles/documentdb-modeling-data)
* [Partitioning data in DocumentDB](/documentation/articles/documentdb-partition-data)
* [Request units](http://go.microsoft.com/fwlink/?LinkId=735027)

To learn more about DocumentDB, see the Azure DocumentDB [documentation](https://azure.microsoft.com/documentation/services/documentdb/).