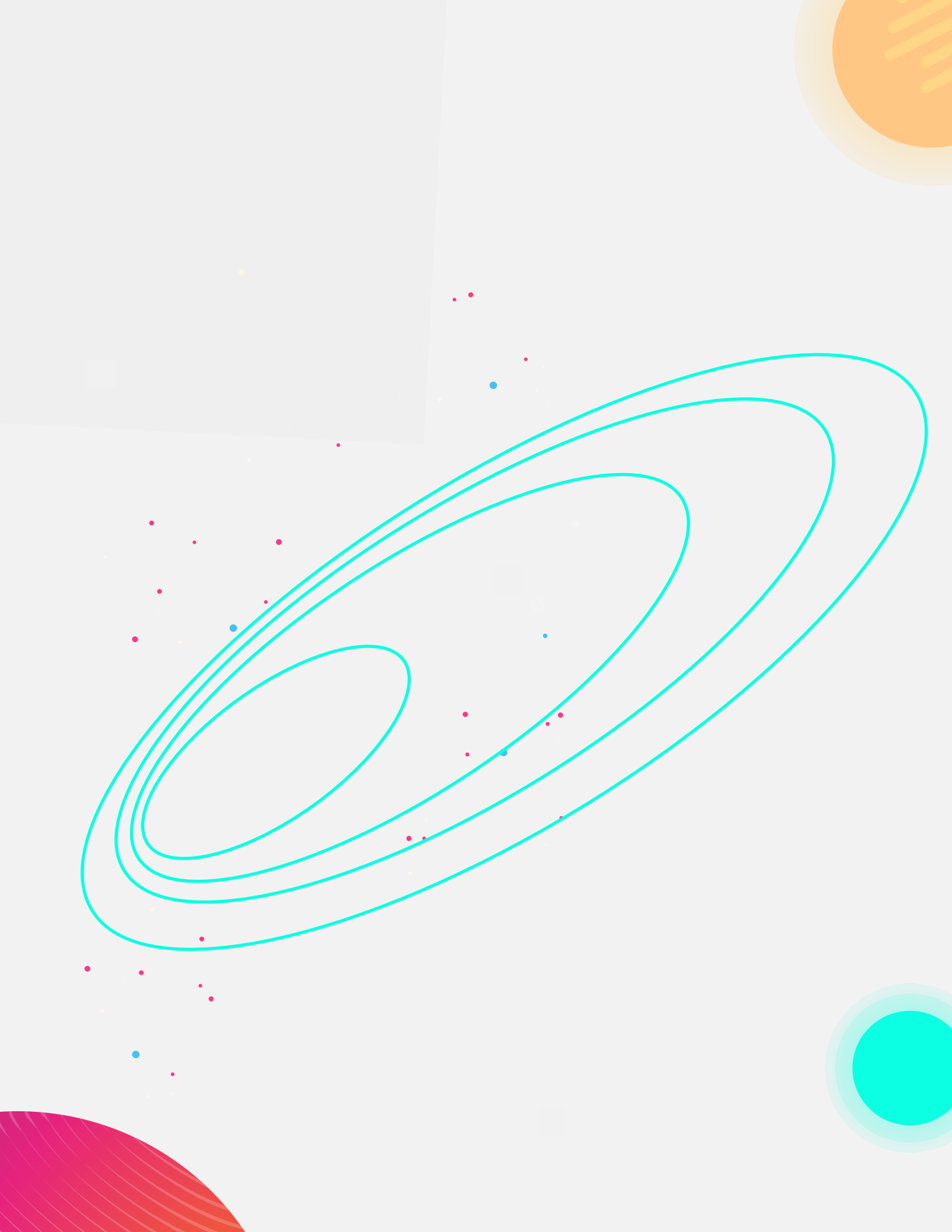


Azure Tips and Tricks: Data

azuredev.tips



Introduction

Hi, folks!



When I reflect back on Azure Tips and Tricks a year ago, I was only thinking that I'd write a couple of posts and move on. Fast-forward to today, the collection has grown to over 150+ tips, as well as videos, conference talks, and now an e-book spanning the entire universe of the Azure platform. What you are currently reading is a special collection of tips based on page views of the entire series over the last year. Before we dive in, you'll notice my pixelated form as you turn each page.

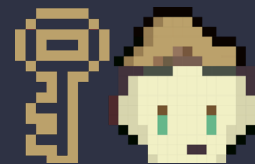
These represent:



Something I found interesting and you may too.



Additional resources to get the most out of this tip.



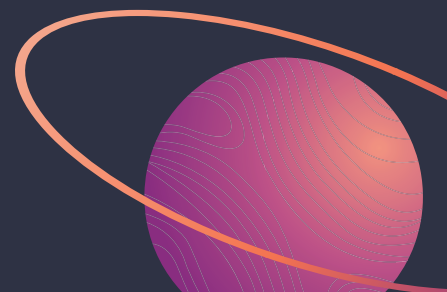
A key takeaway from the tip.

You can stay up to date with the latest Azure Tips and Tricks at:

- Blog - azuredev.tips
- Videos - videos.azuredev.tips
- eBook - ebook.azuredev.tips
- Survey - survey.azuredev.tips

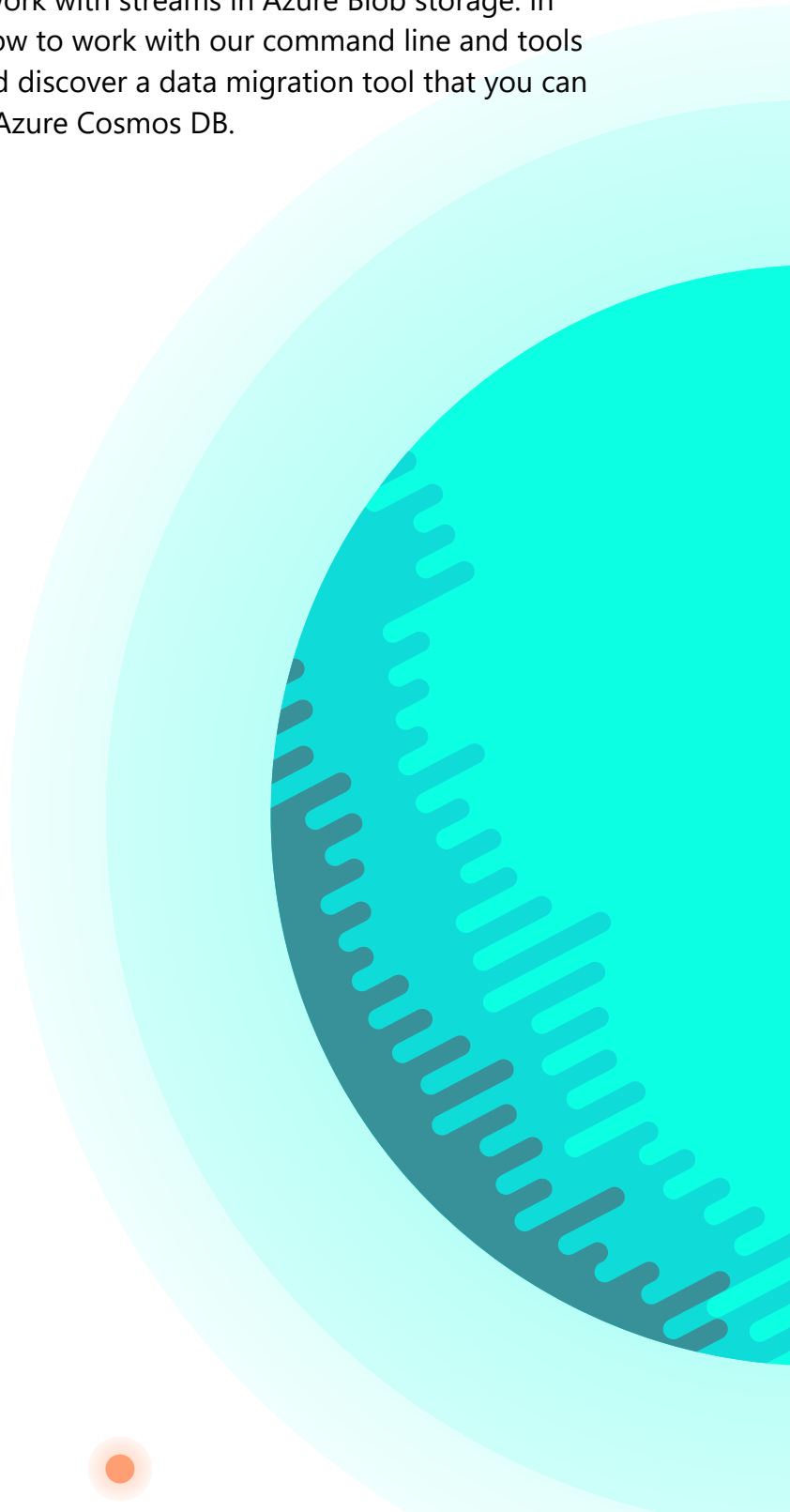
I hope you enjoy reading the eBook as much as I did writing it.

Thanks,
Michael Crump (@[mbcrump](https://twitter.com/mbcrimp))



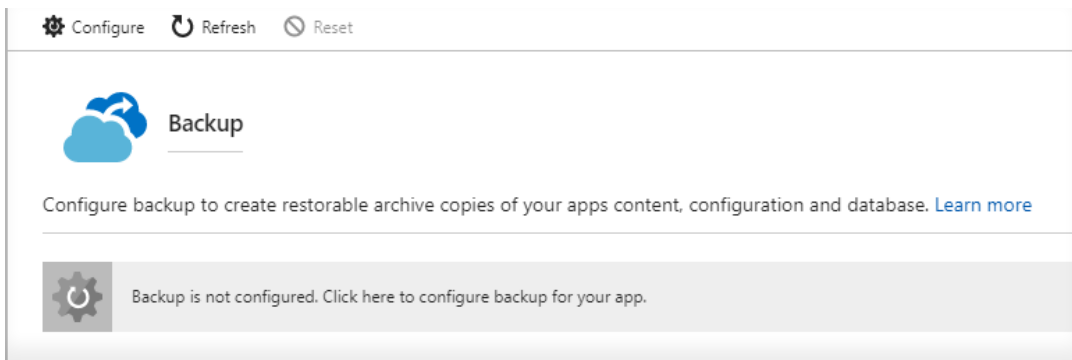
Data

Data and application development go hand in hand. It's no wonder that these four data tips were the best of all time for developers. In this e-book, we'll discover an easy way to configure backups to create copies of your content, configuration, and database. We'll also take a look at how to work with streams in Azure Blob storage. In addition, you'll learn how to work with our command line and tools with Azure Storage, and discover a data migration tool that you can use to move data into Azure Cosmos DB.

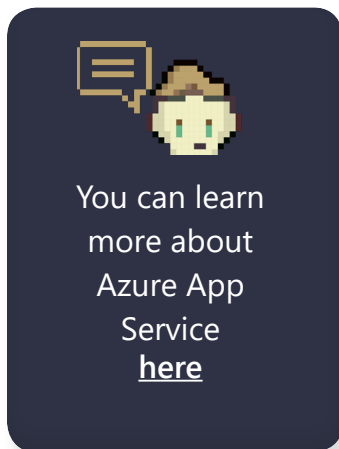


Configure a Backup for your Azure App Service and Database

Most folks don't realize how easy it is to configure a backup copy of your Azure App Service to ensure you have restorable archive copies of your app and database. In order to take advantage of this, you'll need to log into your Azure account and go to your App Service that you created. Look under [Settings](#) and you will see [Backup](#).



Open it and select [Configure](#) and you'll see the following screen:



Backup Storage

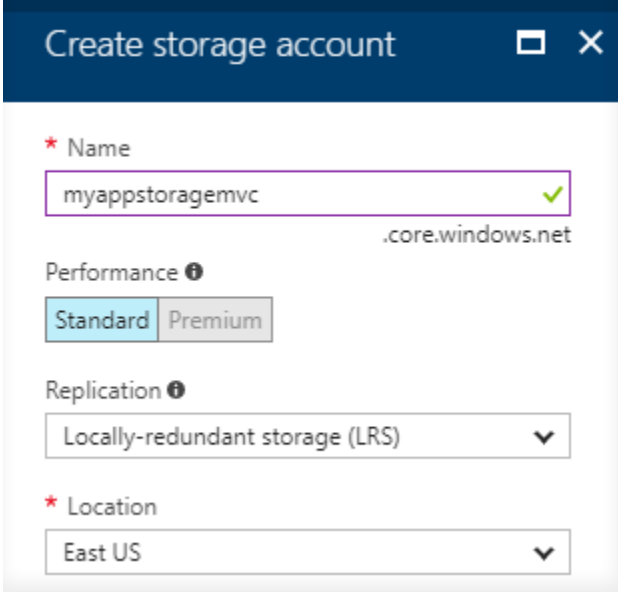
Select the target container to store your app backup.

Storage Settings

Storage not configured

Backup Schedule

Backup Database



Create storage account

* Name
myappstoragemvc ✓
.core.windows.net

Performance ⓘ
Standard Premium

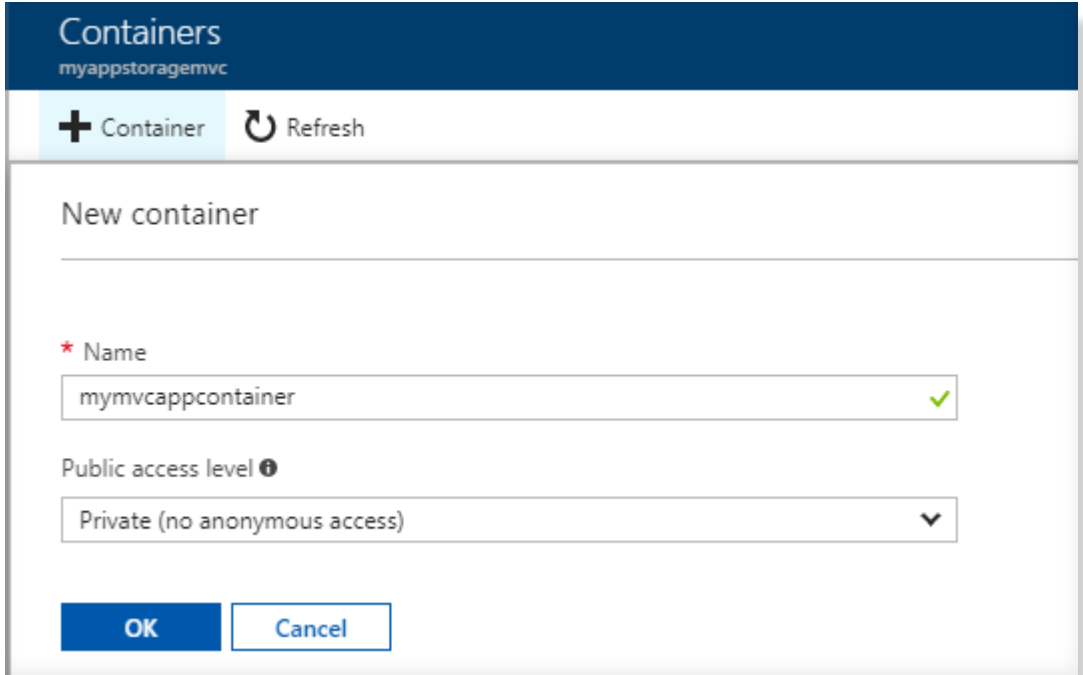
Replication ⓘ
Locally-redundant storage (LRS) ▼

* Location
East US ▼

I provided a Name, selected the Standard under the Performance option, and used Locally-Redundant Storage (LRS) for Replication. For the location field, please use the closest location nearest you.



Quick Tip You can type **help** from the console window for a list of available commands.



Containers
myappstoragemvc

+ Container Refresh

New container

* Name
mymvcappcontainer ✓

Public access level ⓘ
Private (no anonymous access) ▼

OK Cancel



Once completed, you can click on the backup and see a feature called **Snapshot** which automatically creates periodic restore points of your app when hosted in a Premium App Service plan. You can even download a zip of the app.

Next, you'll want to make sure that **Scheduled backup** is set to **On**. You'll want to configure the Days and Hours and then the current schedule that it should back up from. I set mine to back up every **seven** days, and starting from now. You'll also want to set the retention and by default it will keep at least one backup. If you have a database, then you can also add it with just a checkmark.

Once everything is set, you can see that the next backup is configured and can either force it manually or restore from an existing backup with just a visit to the Azure Portal. You typically want to use "manual" restore when you want to look at your backup at the current point in time vs "restore" a backup at a different time that occurred in the past.



Snapshot (Preview)

Snapshots automatically create periodic restore points of your app when hosted in a Premium App Service plan.

Backup Id

3077

Status

Created

Created Time

Wednesday, October 4, 2017, 10:47:27 PM PDT

Download Backup Zip

Zip contains the backup of an app

[Download](#) 



Size Of Backup

0MB





Learn more about
Azure Cosmos DB
[here](#)

Using the Data Migration Tool with Cosmos DB

Migrating data from one format to another is a common task for application developers (even if it is just for testing). I was recently building out an API and needed to dump some data into Cosmos DB. The tool that made short work of this was the [Azure DocumentDB Data Migration Tool](#). In my case, I needed to dump a large JSON file into Cosmos DB. Here is how I did it.



Grab whatever sample file that you'd like to experiment with. For this exercise, I selected a file that is public domain and contains a large set of data.

I'm using the [en_kjv.json](#) JSON file from [here](#)

Now we're ready to begin work!

The Tools + Public Domain Sample Data

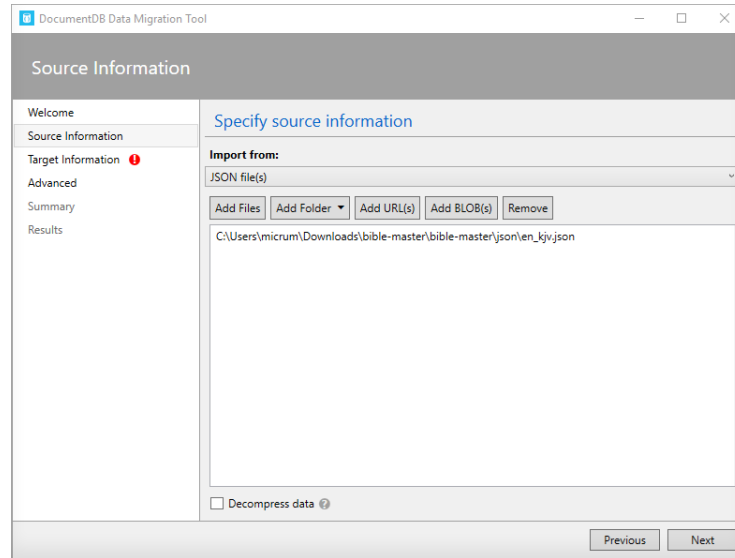
Get to work

Ensure you have a Cosmos DB database id and collection. You can learn how to create a Cosmos DB by going to <https://docs.microsoft.com/en-us/azure/cosmos-db/>. I'm using the following:

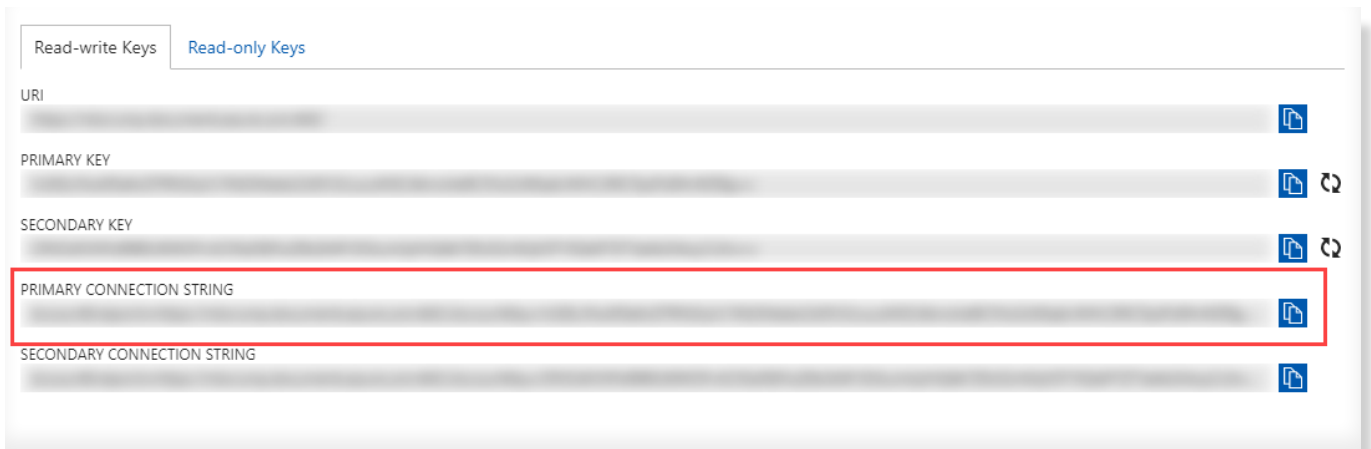
The screenshot shows the 'Add Collection' dialog box in the Azure portal. It contains the following fields and options:

- Database id:** A text input field containing the value 'bible'.
- Collection Id:** A text input field containing the value 'verses'.
- Storage capacity:** Two radio button options: 'Fixed (10 GB)' (which is selected) and 'Unlimited'.
- Throughput (400 - 10,000 RU/s):** A text input field containing the value '400', with minus and plus buttons for adjustment.
- Estimated spend (USD):** A line of text showing '\$0.032 hourly / \$0.77 daily'.
- Footer note:** 'Choose unlimited storage capacity for more than 10,000 RU/s.'

Open the Data Migration Tool and under **Source Information**, point to the local JSON file as shown below.



Go to **Keys** (inside your Cosmos DB blade in the portal) to copy the **Primary Connection String**.

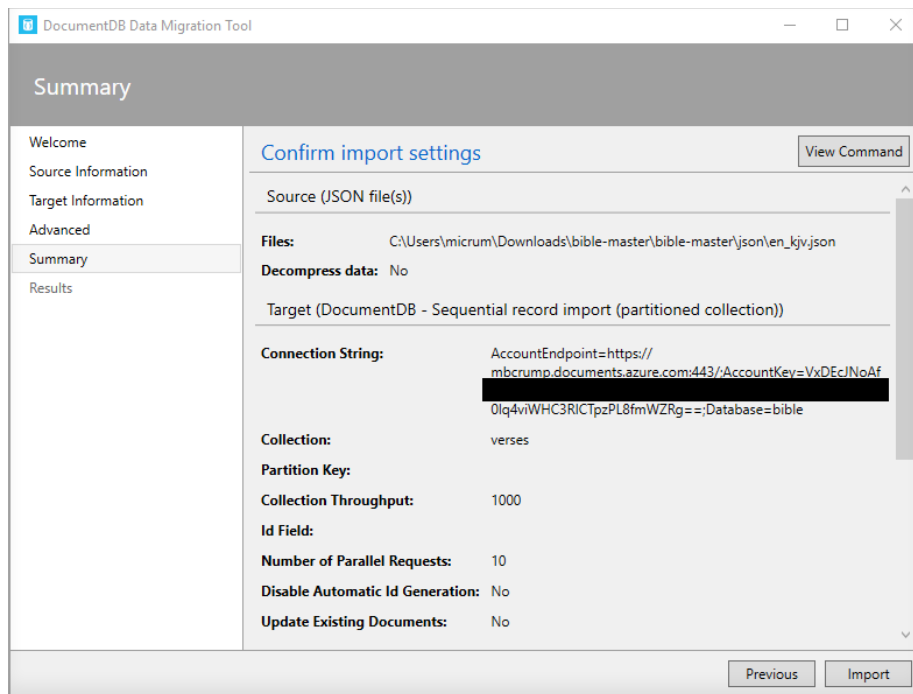


You'll need to append the Database name to the end of the string. For example: **Database=bible** will be appended to the string **AccountEndpoint=https://mbcrump.documents.azure.com:443/;Account-Key=VxDEcJblah=;Database=bible** that I copied out of the portal. Now press **Verify Connection**.

Give it a couple of minutes until you see that it has completed pulling down your code from Git and then go to the new URL of your site. You can find the URL on your overview page. In my case it is, <http://myquizapplication-staging.azurewebsites.net/>

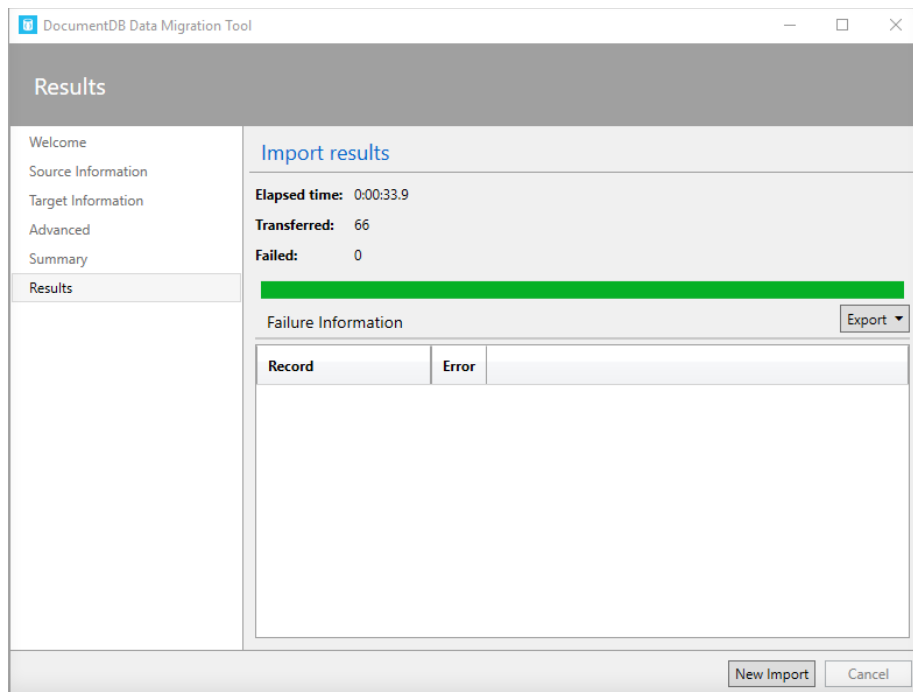
The screenshot shows the 'DocumentDB Data Migration Tool' window. The left sidebar contains a navigation menu with the following items: 'Welcome', 'Source Information', 'Target Information' (which is highlighted with a red exclamation mark icon), 'Advanced', 'Summary', and 'Results'. The main area is titled 'Specify target information'. It includes a dropdown menu for 'Export to:' set to 'DocumentDB - Sequential record import (partitioned collection)'. Below this is a 'Connection String' field with a help icon, containing the text 'AccountEndpoint=https://mbcrump.documents.azure.com:443/;AccountKey=VxDEc/NoAfSaKx'. A 'Verify' button is next to the field. A 'Verify Connection' dialog box is open in the foreground, displaying an information icon and the message 'Successfully connected to DocumentDB account', with an 'OK' button at the bottom. At the bottom of the main window, there are 'Previous' and 'Next' buttons.

You'll need to add the **Collection** and in my case it is **verses**. We'll take the defaults on the next two screens and you'll finally see a **Confirm inport settings** page.



You can even click on [View Command](#) to see the command that will be used to migrate your data. This is helpful to just learn the syntax.

You'll finally see the Import has completed with 66 transferred.



If you go back to the Azure Portal, open Cosmos DB, and look under [Data Explorer](#), you'll see the data has been imported successfully into our collection.

New Collection New SQL Query New Stored Procedure New UDF New Trigger Delete Collection Delete Database Feedback

COLLECTIONS

- bible
 - verses
 - Documents
 - Scale & Settings
 - Stored Procedures
 - User Defined Functions
 - Triggers

Documents

SELECT * FROM c Edit Filter

id
af49605b-7237-473...
83d70092-5521-46...
62fea013-68b3-4e2...
2a7bd04d-e82e-46...
1e65b742-bca6-44...
80fca2c0-6d39-4ea...
e8f1e498-e900-4eb...
8867e8e7-66c9-41...
f8b62771-3ea6-419...
f802093d-1de9-4ad...
1992ae93-9b91-46...
83a647c3-f409-42e...
7eed370-4efc-4a4...
6a532a09-a974-43...
07c4c93-94db-48a...
60ebb68d-f01b-41...
383a96e1-63db-47...
ec492d0a-ca41-4a9...
08e48086-1fe4-447...
2290454f-e5a1-482...
26daedec-029e-46...
10a5ahf0-f47a-457...

Load more

```

1 {
2   "abbrev": "ex",
3   "book": "Exodus",
4   "chapters": [
5     {
6       "1": {
7         "1": "Now these {are} the names of the children of Israel, v
8         "2": "Reuben, Simeon, Levi, and Judah,",
9         "3": "Issachar, Zebulun, and Benjamin,",
10        "4": "Dan, and Naphtali, Gad, and Asher.",
11        "5": "And all the souls that came out of the loins of Jacob
12        "6": "And Joseph died, and all his brethren, and all that ge
13        "7": "And the children of Israel were fruitful, and increas
14        "8": "Now there arose up a new king over Egypt, which knew r
15        "9": "And he said unto his people, Behold, the people of the
16        "10": "Come on, let us deal wisely with them; lest they mult
17        "11": "Therefore they did set over them taskmasters to affli
18        "12": "But the more they afflicted them, the more they multi
19        "13": "And the Egyptians made the children of Israel to serv
20        "14": "And they made their lives bitter with hard bondage, i
21        "15": "And the king of Egypt spake to the Hebrew midwives, o
22        "16": "And he said, When ye do the office of a midwife to th
23        "17": "But the midwives feared God, and did not as the king
24        "18": "And the king of Egypt called for the midwives, and sa
25        "19": "And the midwives said unto Pharaoh, Because the Hebre
26        "20": "Therefore God dealt well with the midwives: and the p
27        "21": "And it came to pass, because the midwives feared God,
28        "22": "And Pharaoh charged all his people, saying, Every sor
29      },
30    },
31    {
32      "2": {
33        "1": "And there went a man of the house of Levi, and took {t
34        "2": "And the woman conceived, and bare a son: and when she
35        "3": "And when she could not longer hide him, she took for h
36        "4": "And his sister stood afar off, to wit what would be d
37        "5": "And the daughter of Pharaoh came down to wash {herself
38        "6": "And when she had opened {it}, she saw the child: and,
39        "7": "Then said his sister to Pharaoh&#x27;s daughter, Shall
40        "8": "And Pharaoh&#x27;s daughter said to her, Go. And the
  
```



Learn more about
our variety of data
options
[here](#)

Uploading and Downloading a Stream into an Azure Storage Blob

Azure Storage is described as a service that provides storage that is available, secure, durable, scalable, and redundant. Azure Storage consists of 1) Blob storage, 2) File Storage, and 3) Queue storage. In this tip, we'll take a look at how to upload and download a stream into an Azure Storage Blob with C#.



If you need help setting up a project for the code below then go [here](#)

Upload a File

Now that we've created the Azure Storage Blob Container, we'll upload a file to it. We'll build off our last code snippet and add the following lines of code to upload a file off our local hard disk:

```
static void Main(string[] args)
{
    var storageAccount =
        CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("StorageConnection"));
    var myClient = storageAccount.CreateCloudBlobClient();
    var container = myClient.GetContainerReference("images-backup");
    container.CreateIfNotExists(BlobContainerPublicAccessType.Blob);

    //lines modified
    var blockBlob = container.GetBlockBlobReference("mikepic.png");
    using (var fileStream = System.IO.File.OpenRead(@"c:\mikepic.png"))
    {
        blockBlob.UploadFromStream(fileStream);
    }
    //lines modified

    Console.ReadLine();
}
```

If we switch over to our Storage Account and navigate inside the container, we'll see our new file has been added:

images-backup
Container

Upload

Refresh

Delete container

Container properties

Access policy

Location: images-backup

Search blobs by prefix (case-sensitive)

NAME	MODIFIED	BLOB TYPE	SIZE	LEASE STATE
 mikepic.png	1/1/2018, 12:34:21 PM	Block blob	182.88 KiB	Available ...

Download a File

Now that we've uploaded a file to the Azure Storage Blob Container, we'll download a file from it.

We'll build off our last code snippet and add the following lines of code to download a file from our local hard disk and give it new name:

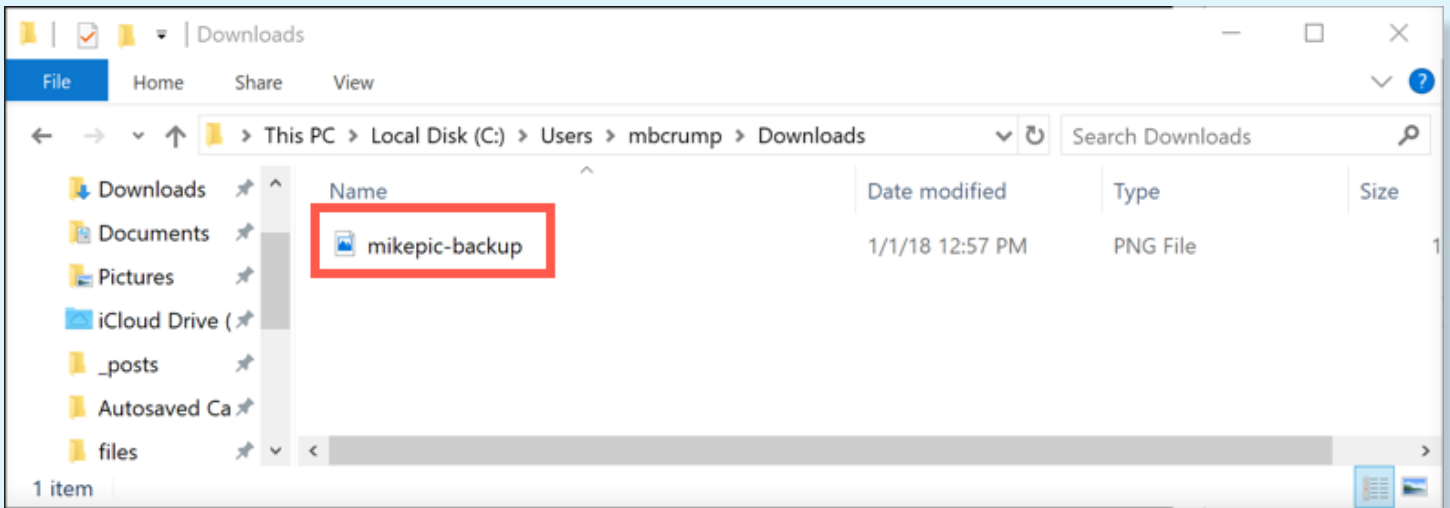
```
static void Main(string[] args)
{
    var storageAccount =
CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("StorageConnection"))
};
    var myClient = storageAccount.CreateCloudBlobClient();
    var container = myClient.GetContainerReference("images-backup");
    container.CreateIfNotExists(BlobContainerPublicAccessType.Blob);

    //lines modified
    var blockBlob = container.GetBlockBlobReference("mikepic.png");
    using (var fileStream =
System.IO.File.OpenWrite(@"C:\Users\mbcrump\Downloads\mikepic-backup.png"))
    {
        blockBlob.DownloadToStream(fileStream);
    }
    //lines modified

    Console.ReadLine();
}
```

Note that are now using the [OpenWrite](#) method and specifying a new name. We are also taking advantage of the [DownloadToStream](#) method. If we run the application, our new file should be in the downloads folder.





What is AzCopy?

AzCopy is a command line utility designed for copying data to/from Microsoft Azure Blob, File, and Table storage, using simple commands designed for optimal performance. You can copy data between a file system and a storage account, or between storage accounts. *(courtesy of docs)*



You can download either the latest version of AzCopy on Windows or Linux.

Working with AzCopy and Azure Storage

You can easily work with AzCopy to manipulate Azure Storage containers and more. In this tip, we'll explore AzCopy in the context of Azure Storage containers.

For this example, I'm going to use Windows. After I downloaded and installed the utility, I navigated inside my command prompt to the following folder `%ProgramFiles(x86)%\Microsoft SDKs\Azure\AzCopy` and ran the `azcopy.exe` command to ensure everything was working properly.

```
C:\Windows\System32\cmd.exe

C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>azcopy

-----
AzCopy 7.1.0 Copyright (c) 2017 Microsoft Corp. All Rights Reserved.
-----
# AzCopy is designed for high-performance uploading, downloading, and copying
data to and from Microsoft Azure Blob, File, and Table storage.

# Command Line Usage:
AzCopy /Source:<source> /Dest:<destination> [options]

# Options:
[/SourceKey:] [/DestKey:] [/SourceSAS:] [/DestSAS:] [/V:] [/Z:] [/@:] [/Y]
[/NC:] [/SourceType:] [/DestType:] [/S] [/Pattern:] [/CheckMD5] [/L] [/MT]
[/XN] [/XO] [/A] [/IA] [/XA] [/SyncCopy] [/SetContentType] [/BlobType:]
[/Delimiter:] [/Snapshot] [/PKRS:] [/SplitSize:] [/EntityOperation:]
[/Manifest:] [/PayloadFormat:]

-----
For AzCopy command-line help, type one of the following commands:
# Detailed command-line help for AzCopy      --- AzCopy /?
# Detailed help for any AzCopy option        --- AzCopy /?:SourceKey
# Command line samples                       --- AzCopy /?:Sample
You can learn more about AzCopy at http://aka.ms/azcopy.
-----

C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>
```

You may be wondering if you need to do the device login as we did with the Azure CLI. The answer is no, we'll be using our Azure Storage Access Key.



Getting the Azure Storage Access Key

Go ahead and open the Azure Portal and navigate to the Azure Storage account that we worked with [earlier](#).

Look under **Settings**, then **Access Keys** and copy the key1.

mbcrumpsa - Access keys
Storage account

Search (Ctrl+/,)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

SETTINGS
Access keys
Configuration
Shared access signature

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. [Learn more](#)

Storage account name: mbcrumpsa

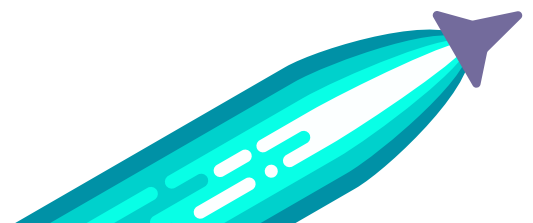
NAME	KEY	CONNECTION STRING
key1	TWZajZlBjqEEoSUKQ+Ail55HNiX0Ux8y9S+KTH34...	DefaultEndpointsProtocol=https;AccountName=...
key2	d/qo8eKSh6MMKosIUQbwHwq4rZnQhz1emyYKQ...	DefaultEndpointsProtocol=https;AccountName=...

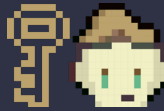
Store the key1 somewhere that you can retrieve it again.

Kick the tires with a couple of commands.

We can easily download a file from our Azure Storage Blob Container that we've been working with by using the following command:

```
AzCopy /Source:https://mbcrumpsa.blob.core.windows.net/images-backup  
/Dest:C:\mytest /SourceKey:thekeyyoucopiedearlier /Pattern:"mikepic.png"
```





Keep in mind: The main difference between these two commands is the use of **SourceKey** for downloading and **DestKey** for uploading. The key that is being used is identical (named key1 from the example above).

We can do the reverse and upload a file from our hard disk to Azure Storage Blob Container with the following command:

```
AzCopy /Source:C:\mytest  
/Dest:https://mbcrumpsa.blob.core.windows.net/images-backup  
/DestKey:thekeyyoucopiedearlier /Pattern:"mikepicnew.png"
```

Finally, you can copy from one Azure Storage account to another one with the following command:

```
AzCopy /Source:https://mbcrumpsa.blob.core.windows.net/images-backup  
/Dest:https://mbcrumpsa.blob.core.windows.net/images  
/SourceKey:thekeyyoucopiedearlier /DestKey:thekeyyoucopiedearlier  
/Pattern:"mikepicnew.png"
```

In this case, I am copying a file named **mikepicnew.png** from **images-backup** to **images** and then I'll refresh the container.

The screenshot displays the Azure Storage Explorer interface. On the left, the 'Essentials' pane shows a list of containers: 'images' (selected) and 'images-backup'. The main pane shows the 'Location: images' container, which contains a single blob named 'mikepic.png'. The blob's details are as follows:

NAME	MODIFIED	BLOB TYPE	SIZE	LEASE STATE
mikepic.png	12/30/2017, 10:36:40 AM	Block blob	182.88 KiB	Available

Below the Storage Explorer, a Windows command prompt window is open, showing the execution of the AzCopy command:

```
C:\Windows\System32\cmd.exe  
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>
```

Conclusion

There are 130+ additional tips waiting on you that cover additional topics such as :

- App Services
- CLI
- Cloud Shell
- Cognitive Services
- Containers
- Cosmos DB
- Functions
- IoT
- Logic Apps
- Portal
- PowerShell
- Productivity
- Storage
- SQL and Search

Find all of these and more at azuredev.tips

Don't forget that if you are modernizing an existing application or building a new app, you can get started Azure for free and get:

- \$200 credit toward use of any Azure service
- 12 months of free services—including compute, storage, network, and database
- 25+ always-free services—including serverless, containers, and artificial intelligence

[Start free](#)

Until next time,

Michael Crump [@mbcrump](#)

signing off...



Azure Tips and Tricks

azuredev.tips



Copyright © 2018 by Microsoft Corporation. All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Made with love By [Red Door Collaborative.com](http://RedDoorCollaborative.com)