Best Practices for Blob Storage

1. Always define content-type of each element

It's crucial to correctly define the content type of each Storage Blob in order for the client to correctly handle the contents being sent.

2. Always define the Cache-Control header for each element

The Cache-Control header is very important as it allows you to improve the availability of a Blob and at the same time decrease significantly the number of transactions that are made in each storage control. For example, imagine you have a static website placed inside the Blob Storage, if the Cache-Control header is configured correctly; the cache can be placed on the client-side in order to decrease the traffic being served as the Blob already exists on the client side.

3. Always upload contents to Blob Storage in parallel

Uploading contents data to Blob Storage can be time consuming. Obviously, performance depends on the volume of data being sent, however it is possible to perform this operation faster by uploading data in parallel, which is supported by both Page and Block Blobs (I'll discuss these below). This way you can reduce the amount of time needed to upload Blob contents. For example, I uploaded 70GB of data to Windows Azure Storage the other day using a third-party tool, and suddenly it was telling me that it would take over 1700 hours to complete! However, when I performed a parallel upload, I was able to upload everything in just approximately 8 hours.

4. Choose the right type of Blob

Windows Azure Blob Storage comes in two varieties: Block and Page Blobs, both with differing characteristics. It is crucial to select the correct type of Blob in order to get the best results. Choose Block Blob if you want to stream your contents, as it can be consumed in blocks, rendering it easier and simpler for streaming solutions; it's also crucial to parallel the upload and downloads of those blobs. Choose Page Blob if you need to write intensively to the Blob; for example a VHD (Cloud Drive is a Page Blob), as Page Blobs allow you to write to a particular part, or 'page' of the Blob. As a result this leaves all other contents unaffected if they are being accessed.

5. Use 'Get Blob Properties' or 'Get Blob Metadata' whenever you only need that specific information

Blobs have distinct information in properties and metadata. Properties are defined by default by Windows Azure, and Metadata are additional properties you can add to the Blob. In order to avoid unnecessary usage of the Blob, use the correct GET method, as well as not getting the entire Blob if we only require its metadata or property information.

6. Take snapshots to improve availability and caching

To increase the availability of Storage Blobs, it's possible to create a snapshot of the Blob. This will allow you to have a kind of 'copy' of the Blob without needing to pay extra for it, as long as the snapshot does not differ from the original. Consequently snapshots may be used to increase the availability of the system since we can have several 'copies' of the same Blob and serve them to the customer. Furthermore, and more importantly snapshots may be used as a way to improve availability by assigning them as the default Storage Blob to be accessed by all clients performing read operations, leaving the original Blob only for writes. Overall snapshots allow the user to perform caching of data at the Blob level, thus increasing the availability of the Blob.

7. Enable the 'Content Delivery Network' for better availability

Another very important part of improving availability and reducing latency is the usage of the Content Delivery Network (CDN). The CDN reduces latency and increases availability by placing a duplicate of your Blob closer to the client. Accordingly each client is redirected to the closest CDN node, of the Blobs. It is important to note that since a copy of the Blob will be placed on a CDN Node closest to the client, the costs will increase, however you will not be charged for Storage transaction costs for each client access to the Blob since the client is hitting the CDN node and not the Storage. This happens because the "copy" already exists in the CDN. But let's explain this more in terms of cost when using CDN. Once you enable CDN your blob will be automatically replicated to all CDN nodes, so there will be costs in terms of Storage Transactions and Transfer (From Storage to each CDN node) since the blob is being downloaded, but after that every client that accesses it will be served with the "copy" that exists in the CDN node closest to him. Only Traffic costs will be charged due to the download. The process will restart once the "copy" that exists in the CDN node expires due to the TTL.

Quick Note: CDN is in Windows Azure Storage and is only available for public containers, so the process will be different for private containers.

8. Serve static contents directly from Blob Storage

Windows Azure Blob Storage is ideal for hosting static websites, since it doesn't require any scaling work in order to improve availability, because this will be done automatically by the platform. It's also possible to reduce your costs if the best practices presented previously are followed. Consequently this is an effective and economical way to host a static website.

9.

Turn on soft delete for blobs Soft delete for blobs enables you to recover blob data after it has been deleted. For more information on soft delete for blobs, see Soft delete for Azure Storage blobs.

Allow Shared Access Signature Tokens Over HTTPS Only

Ensure that Shared Access Signature (SAS) tokens are allowed only over the HTTPS protocol.

Check for Overly Permissive Stored Access Policies

Ensure that Azure Storage shared access signature (SAS) tokens are not using overly permissive access policies.

Check for Publicly Accessible Web Containers

Ensure that Azure Storage containers created to host static websites are not publicly accessible.

Check for Sufficient Soft Deleted Data Retention Period

Ensure there is a sufficient retention period configured for Azure Blob Storage soft deleted data.

Disable Anonymous Access to Blob Containers

Ensure that anonymous access to blob containers is disabled within your Azure Storage account.

Enable Blob Storage Lifecycle Management

Ensure that Azure Blob Storage service has a lifecycle management policy configured.

Enable Immutable Blob Storage

Ensure that critical Azure Blob Storage data is protected from accidental deletion or modification.

Enable Logging for Azure Storage Queue Service

Ensure that detailed storage logging is enabled for the Azure Storage Queue service.

Enable Secure Transfer in Azure Storage

Ensure that "Secure transfer required" security feature is enabled within your Azure Storage account configuration.

Enable Soft Delete for Azure Blob Storage

Ensure that Soft Delete feature is enabled for your Microsoft Azure Storage blob objects.

Enable Trusted Microsoft Services for Storage Account Access

Allow Trusted Microsoft Services to access your Azure Storage account resources.

Expire Shared Access Signature Tokens

Ensure that your Shared Access Signature (SAS) tokens expire within an hour.

Limit Storage Account Access by IP Address

Ensure that Azure Storage account access is limited only to specific IP address(es).

Regenerate Storage Account Access Keys Periodically

Regenerate storage account access keys periodically to help keep your storage account secure.

Restrict Default Network Access for Storage Accounts

Ensure that the default network access rule is set to "Deny" within your Azure Storage account.

Review Storage Accounts with Static Website Configuration

Ensure that Azure Storage Accounts with static website configuration are regularly reviewed (informational).

Use BYOK for Storage Account Encryption

Use customer-managed keys (CMKs) for Microsoft Azure Storage accounts encryption.