Azure blob storage

It is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing a massive amount of unstructured data, such as text or binary data.

Blob storage usages:

- It serves images or documents directly to a browser.
- It stores files for distributed access.
- We can stream video and audio using blob storage.
- Easy writing to log files.
- It stores the data for backup, restore, disaster recovery, and archiving.
- It stores the data for analysis by an on-premises or Azure-hosted service.

Azure blob storage is fundamental for the entire Microsoft Azure because many other Azure services will store the data within a storage account, inside the blob storage, and act upon that data. And every blob should be stored in a container.

Container

The container is more like a folder where different blobs are stored. At the container level, we can define security policies and assign those policies to the container, which will be cascaded to all the blobs under the same container.

A storage account can contain an unlimited number of containers, and each container can contain an unlimited number of blobs up to the maximum limit of storage account size (up to 500 TB).

To refer this blob, once it is placed into a container inside a storage account, we can use the URL, which looks like http://mystorageaccount.blob.core.windows.net/mycontainer/myblob.

Blob storage is based on a flat storage scheme. So you can't create a container within a container. Let's take an example - once we create a container like videos and if we want to differentiate between professional videos and personal videos. Then we can prefix the blob names with personnel for personal videos and professional for professional videos. The blob name will be shown as personal-video1, personal-video2 for personal videos, and for professional videos - professional-video1, professional-video2. Like this, we can create a virtual hierarchy, but we can't create a container within a container inside the Azure blob storage service.

Blob Types:

Azure offers three types of blob service:

- **Block blob:** It stores text binary data up-to about 4.7 TB. It is the block of data that can be managed individually. We can use block blobs mainly to improve the upload-time when we are uploading the blob data into Azure. When we upload any video files, media files, or any documents. We can generally use block blobs unless they are log files.
- **Append blob:** It is made up of blocks like block blobs, but are optimized for append operations. It is ideal for an application like logging data from virtual machines. For example - application log, event log where you need to append the data to the end of the file. So when we are uploading a blob into a container using the Azure portal or using code, we can specify the blob type at that time.
- **Page blob:** It stores random access files up-to 8 TB. Page blobs store the VHD files that backs VMs.

Most of the time, we operate with block blob and append blobs. Page blobs are created by default. When we create a virtual machine, the storage account gets created, and the disks associated with the virtual machine will be stored in the storage account. But for most of the storage solutions like we know, we are developing an application like YouTube, or we are developing a monitoring application, in that case, either we use block blobs or append blobs based on the requirement.

Naming and Referencing

The names of container and blob should adhere to some rules. Because the container name and blob name will be a part of the URL when you are trying to access them. They need to adhere to some rules which are specified below.

Container Names

- The name of containers must start with a letter or a number, and can contain only letters, numbers, and the dash (-) character.
- All the letters in a container name must be in lowercase.
- Container names must be 3 to 63 characters long.

Blob Names

- The name of blobs can contain any combination of characters.

- The name of blobs must be at least one character long and cannot be more than 1024 characters long.
- The Azure Storage emulator supports blob names up-to 256 characters long.
- The name of the blobs is case-sensitive.
- The reserved URL characters must be escaped properly.

Metadata & Snapshots

We can store some amount of information against a container or blob as metadata. It is a name-value pair associated with the container or blob. Metadata names must adhere to the name rules for C# identifiers. For example - when we are developing any video streaming application with backend as Azure blob storage, then in that case, when the user uploads a video, we want to store the user information as metadata against that video. It is very useful once we start developing an application based on blob storage.

Blob Snapshots

Snapshot is a read-only version of the blob storage. We can use snapshots to create a backup or checkpoint of a blob. A snapshot blob name includes the base blob URL plus a date-time value that indicates the time when the snapshot was created. Again if we are developing a YouTube-like application and want to retain the previous version of the video, then we can take a snapshot of it and store it once the user updates the video. So, a user like SharePoint can see the previous version of the video and the current version of the video.

To access the snapshot, we have to add a query string at the end of the URL. And a snapshot with a similar date and time when the snapshot was created.

Storage account and Blob service configuration

The first key configuration area is related to the network, which is a storage firewall and virtual networks. Every storage account in Azure has its storage firewall. Within the firewall, we can configure the following rules.

- A set of rules that we can configure is to allow connection from a specific virtual network. If we have an Azure virtual network, we can configure it here to enable the connections from the workloads.

- The second area of a rule is IP address ranges. We can specify an IP address range from where we won't allow the connections to the storage account to access the data.
- The third one is enabling connections from certain Azure services. So, we can specify exceptions in such a way that the connections from trusted Azure services are allowed.

We need to remember that there is a storage firewall associated with the storage account in which we can configure three types of rules.

- We can specify the virtual networks from which the connections are allowed.
- We can specify allowed IP range from where the connections are allowed.
- We can define some exceptions.

Custom Domain

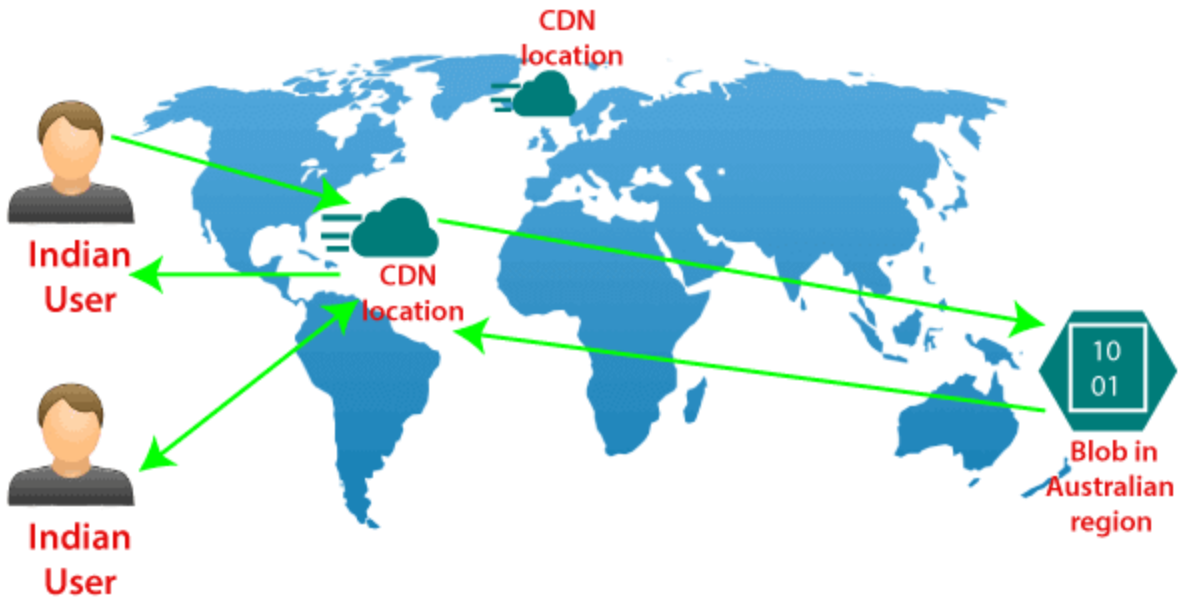
We can configure a custom domain for accessing the block data in our storage account. The default endpoint will be the storage account name "*.blob.core.windows.net*". But in place of that, we can have our domain for the default storage account URL. We can configure our custom domain also. We need to specify our custom domain as "*customdomain/container/myblob*" to access the specific blob.

There are two fundamental limitations that we need to understand when we are using custom domains

- All Azure storage does not natively support HTTPS with the custom domains. We can currently use Azure CDN access blobs by using custom domains over HTTPS.
- Storage accounts currently support only one custom domain name per account. So we can use only one custom domain for all the services within that storage account.

Content delivery network

The Azure content delivery Network (CDN) caches static content at strategically placed locations to provide maximum throughput for delivering content to users. So the most crucial advantage of CDN is providing the content to the users in the most optimal way. So let's see how this works.



We are assuming that we have the blob storage located in the Australia region. So we have most of the users in North India and South India. In that case, we can configure a CDN profile for North India and South India. For example - let's say a North Indian user is trying to access our blob located in the Australia region. So first of all, the request goes to the CDN location. And from the CDN location, the request will further go to blob in the Australian region. For the first user, the block content will be copied to the CDN location, and then eventually delivered to North Indian users. However, when the next North Indian user tries to access that block, they will be redirected to CDN location, and the content will directly be delivered to them from that location in North India itself because the block content is already cached in CDN location.

So from the second user onwards, the content delivery latency is significantly reduced.

Other Configuration areas:

There are some different configuration areas such as performance tier, Access tier, replication strategy, secure transport required, etc.