

# TLS Proxy: Custom domain support for Azure Blob Storage using Application Gateway

## Contents

Overview .....	1
Pre-requisites .....	1
Solution - Overview.....	2
Solution - Detailed Process .....	3
Creating a Storage Account and verifying blob's access.....	3
Configurations of Application Gateway .....	5
Configuration of Storage client and file access.....	8
Callouts/Caveats .....	11
ETA .....	11

## Overview

To use the custom domain feature of Azure Storage one needs to use a public domain since the domain verification process is based on public DNS resolution. This prohibits the use of internal domain name like storage.contoso.corp. Moreover, the Azure Storage supports use of HTTPS for custom domains through Azure CDN only. The Azure CDN is also required when using root domain (e.g. contoso.com) for a custom domain.

With Application Gateway fronting a storage as a backend, we can use internal custom domain, with easier certificate management for TLS termination as well as use of primary domain.

In this document, we will look at setting up an Azure Storage Blob Container and access it using end-to-end TLS through the Application Gateway. The Azure Storage will be with Private Endpoint enabled.

## Pre-requisites

1. Storage Account with a Blob Container
2. Application Gateway using L7 capabilities (HTTPS)
3. SSL certificate for the custom domain pointing to Application Gateway
4. MS Azure Storage Explorer client

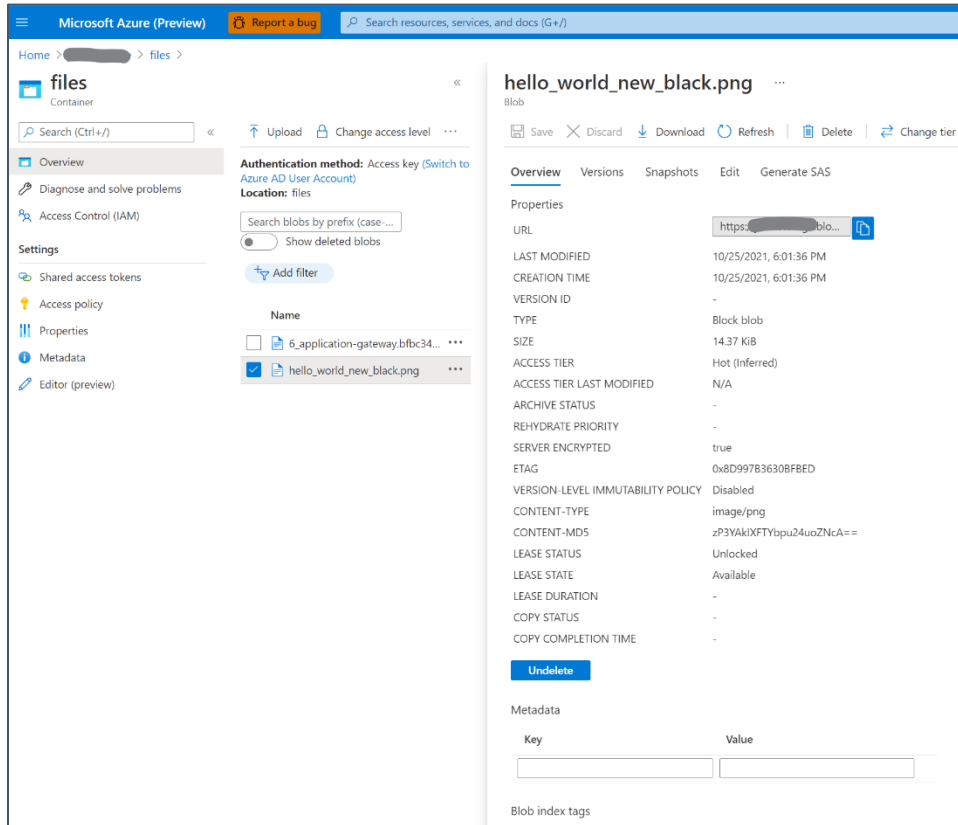
## Solution - Overview

1. Create Storage account, create container and uploaded a JPG file. Verify that image's accessibility in a browser using the default domain name and via public internet.
2. Now enable Private Endpoint feature on Storage resource.
3. Assuming an Application Gateway resource already exists, its configuration will need to be updated as described below:
  - a. Backend Target – Use the default FQDN of storage
  - b. For HTTP settings,
    - i. Choose backend protocol as "HTTPS"
    - ii. Set "Use well known CA certificate" to Yes
    - iii. Set "Override with specific domain name" to specify the default FQDN of Storage or use "Pick host name from backend target", if you are using FQDN for backend.
  - c. For Health Probe, use
    - i. HTTPS
    - ii. Pick hostname from HTTP setting
    - iii. Pick port from HTTP setting
    - iv. Path - as any file in the Blob container
4. Now configure the custom private domain to point to the Application Gateway's frontend IP using an A record.
5. Verify using a browser client.
6. Verify using MS Azure Storage Explorer client

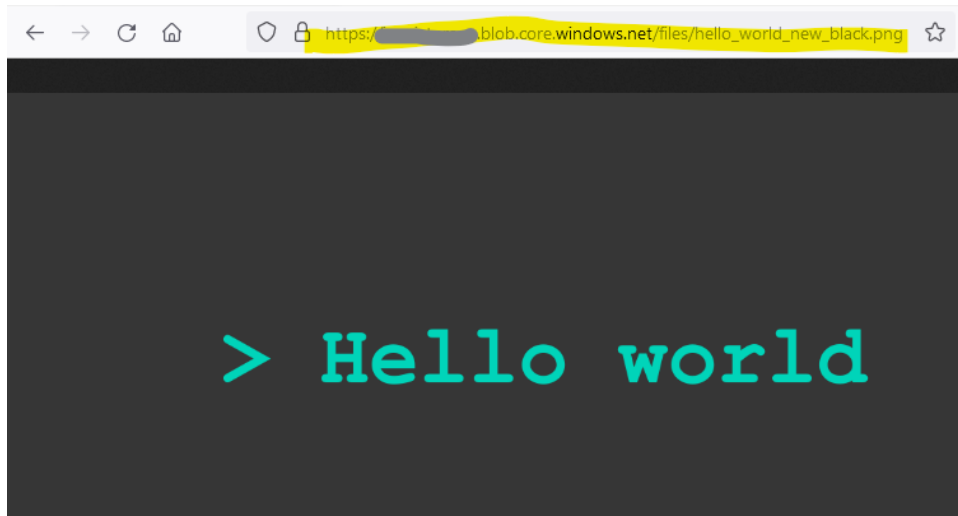
## Solution - Detailed Process

### Creating a Storage Account and verifying blob's access

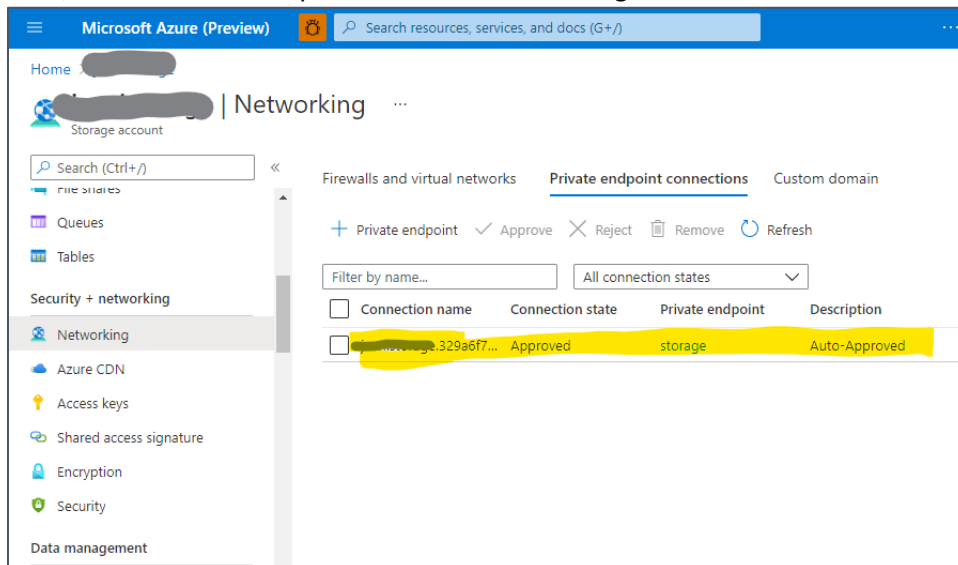
1. Create Storage account, create container and uploaded a JPG file.



Copy the file URL from Properties section and resolve it in a browser to confirm if this basic configuration is working fine. The URL will be of the form <https://<account-name>.blob.core.windows.net/files/<file-name>>



2. Now, enable Private Endpoint feature for the Storage resource.



This [documentation](#) describes the process to enable a Private Endpoint while creating Storage resource itself. You can also perform these steps after its creation via its Networking Blade.

## Configurations of Application Gateway

3. Configure the Application Gateway in the described manner.
  - a. Backend Target – Use the default FQDN of the Storage.

### Edit backend pool ...

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.

Name

blob-backend

Add backend pool without targets

☐ Yes ☒ No

Backend targets

1 item

Target type	Target	
IP address or FQDN	john@blob.blob.core.windows.net	...
<input type="text" value="IP address or FQDN"/>	<input type="text"/>	

Associated rule

[blob-rule](#)

b. For HTTP settings

### Add HTTP setting

Backend protocol  
☐ HTTP ☒ HTTPS

Backend port \*

Trusted root certificate  
For end-to-end SSL encryption, the backends must be in the allowlist of the application gateway. Upload the public certificate of the backend servers to this HTTP setting.

Use well known CA certificate  
☒ Yes ☐ No

Additional settings

Cookie-based affinity ⓘ  
☐ Enable ☒ Disable

Connection draining ⓘ  
☐ Enable ☒ Disable

Request time-out (seconds) \* ⓘ

Override backend path ⓘ

Host name  
By default, Application Gateway does not change the incoming HTTP host header from the client and sends the header unaltered to the backend. Multi-tenant services like App service or API management rely on a specific host header or SNI extension to resolve to the correct endpoint. Change these settings to overwrite the incoming HTTP host header.

Override with new host name  
☒ Yes ☐ No

Host name override  
☐ Pick host name from backend target  
☒ Override with specific domain name

Use custom probe ⓘ  
☒ Yes ☐ No

Custom probe \*

- c. For Health Probe, use

**customprobe\_for\_blob**

Name customprobe\_for\_blob

Protocol \* ☐ HTTP ☒ HTTPS

Pick host name from backend HTTP settings ☒ Yes ☐ No

Pick port from backend HTTP settings ☒ Yes ☐ No

Path \*

Interval (seconds) \*

Timeout (seconds) \*

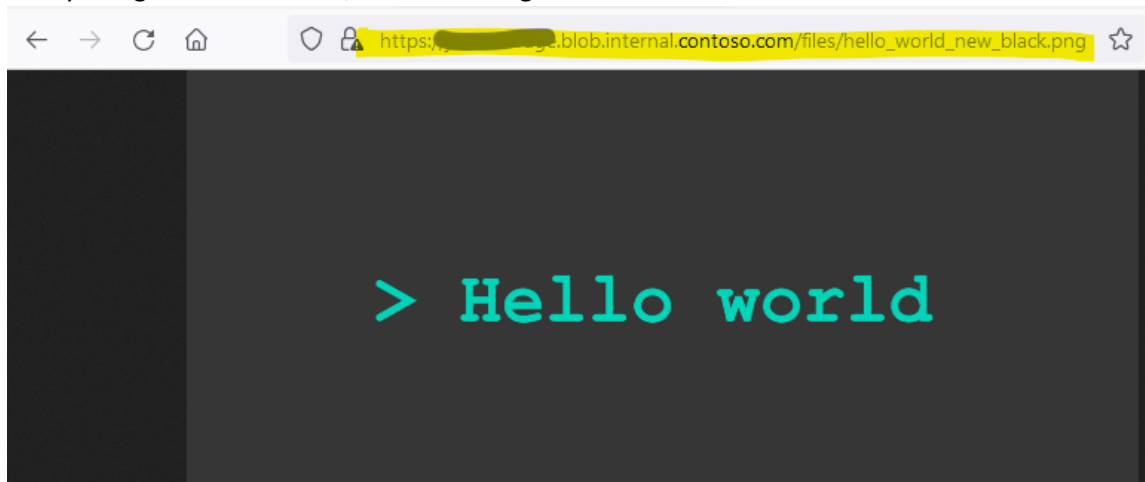
Unhealthy threshold \*

Use probe matching conditions ☐ Yes ☒ No

HTTP settings

Before we proceed, ensure the custom domain's DNS resolves to Application Gateway IP.

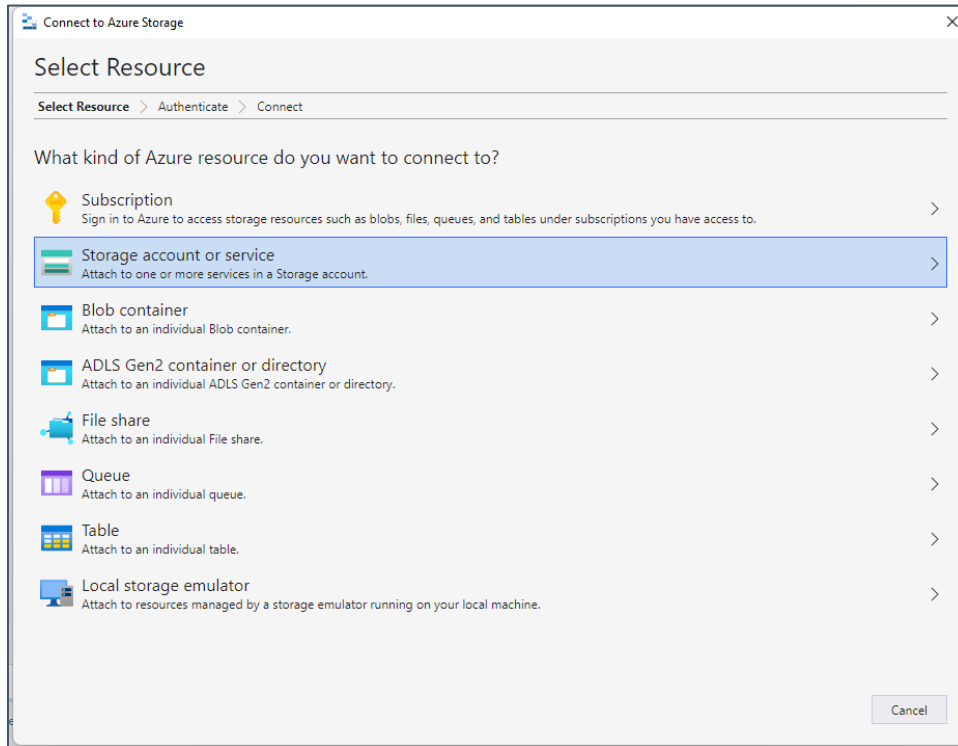
4. Verify using a browser client, this time using custom domain in the URL.



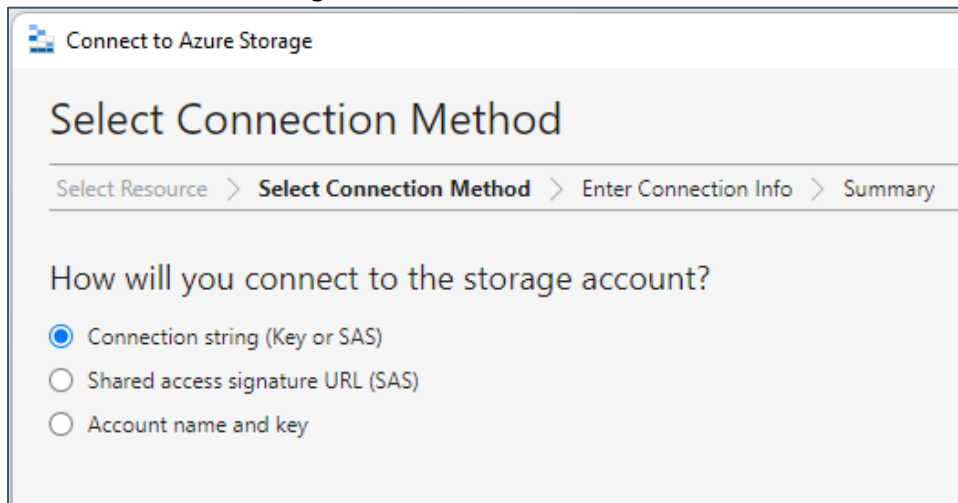
## Configuration of Storage client and file access

### 5. Verify using MS Storage Explorer client.

Click the add connection button to open the below screen. Select “Storage account or service”.



Choose “Connection string”.



Go back to Azure Storage resource in the portal and open “Access keys” blade to copy the given connection string.



Home > jsonistorage

Storage account

## Access keys

Search (Ctrl+/) << Show keys Set rotation reminder Refresh

Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location, such as Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.

Remember to update the keys with any Azure resources and apps that use this storage account. [Learn more](#)

Storage account name: [redacted]

**key1**  
Last rotated: 10/25/2021 (17 days ago)  
Rotate key  
Key: [redacted]  
**Connection string**: [redacted]

**key2**  
Last rotated: 10/25/2021 (17 days ago)  
Rotate key  
Key: [redacted]

**Navigation Menu:**

- Security + networking
  - Networking
  - Azure CDN
  - Access keys**
  - Shared access signature
  - Encryption
  - Security
- Data management
  - Geo-replication
  - Data protection
  - Object replication
  - Blob inventory
  - Static website
  - Lifecycle management
  - Azure search

In the next step of Storage connection configuration, add the copied connection string. Please use only the custom domain part without "blob" or storage account name in it.

Connect to Azure Storage

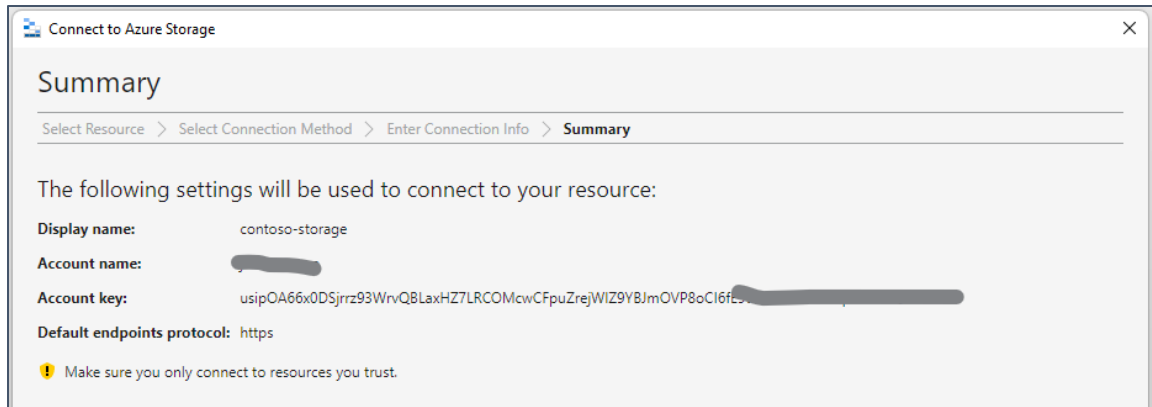
### Enter Connection Info

Select Resource > Select Connection Method > **Enter Connection Info** > Summary

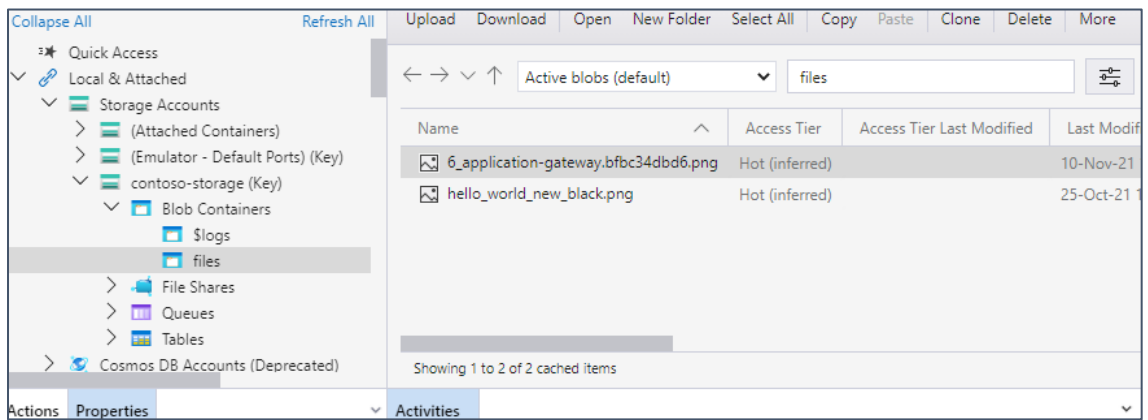
Display name: contoso-storage

Connection string:

DefaultEndpointsProtocol=https;AccountName=[redacted];AccountKey=usipOA66x0DSjrrz93WrvQBLaXHZ7LRCoMwCFpuZrejWIZ9YBjImOVP8oCI6fE9tQoamAdH...;EndpointSuffix=internal.contoso.com



That's about it!



## Callouts/Caveats

1. The [Custom Domain for Azure Storage](#) need NOT be configured in Azure Storage Account. The desired custom domain can simply point to Application Gateway's IP via an A record.
2. The "EndpointSuffix" in the Connection String for Storage Explorer must include only custom domain part. Taking example of Blob containers, the Storage Explorer will form a connection URL as <https://<storage-account>.blob.<custom-domain>>. Since "<account name>" and "blob" are auto added, you must include only the precise custom domain in the connection string endpoint suffix.

If the Account Name = mystorage and Custom Domain = shop.contoso.com, the string will look as

DefaultEndpointsProtocol=https;AccountName=mystorage;AccountKey=<key>==;EndpointSuffix=shop.contoso.com;

3. This document assumes that the Private DNS is configured in the Application Gateway virtual network. This Private DNS contains the required A record for <accountname>.privatelink.blob.core.windows.net
4. While configuring Private Link from clients other than portal, the subnet in which the endpoint would reside must have "privateLinkServiceNetworkPolicies" property disabled. More on this [here](#).
5. Due to current limitation with Application Gateway, any PaaS service's default FQDN should be added to the backend pool **after** configuring the private link, otherwise the default FQDN continues to point to its public IP. [ETA for this fix: Early 2022. As a workaround, you can even perform any PUT operation on gateway which will refresh its DNS.]
6. This document assumes that the Listener is created with HTTPS protocol and an appropriate certificate.

## ETA

Private Preview: 15 Feb, 2022

-----End of File-----