

## TABLE CONTENTS

● Introduction .....	2
● Benefits of hosting static website on Azure .....	2
● Prerequisites .....	3
● Outline of workflow .....	4
● Getting Started	
○ Create a Resource Group .....	5
○ Create a Storage Account .....	7
● Enable static website on Storage Account .....	10
● Upload web contents to container	
○ Downloading AzCopy .....	13
○ Authorize AzCopy to access Storage Account .....	14
○ Running AzCopy script .....	21
● Configure Azure CDN .....	24
● Configure a custom domain to Azure CDN endpoint	
○ Domain registered on Azure .....	28
○ Domain registered with another domain registrar .....	37
● Resources .....	44
● Conclusion .....	45

# HOSTING A STATIC WEBSITE ON AZURE STORAGE WITH AZURE CDN

## INTRODUCTION

A colleague reached out to me some time ago and wanted to host a static website on Azure storage. We haven't had time to go through the steps and achieve his goal which is to host a static website on Azure.

Now I made this demo for those who would like to host a static website on Azure cloud, specifically into an Azure Storage Account using the Azure portal. I would make this demo a step by step walk through so even you that are new to the cloud can easily host your static website on Azure Storage.

## WHAT ARE THE BENEFITS OF HOSTING YOUR STATIC WEBSITE ON AZURE?

Okay so why should you host your static website on Azure? there are good reasons for wanting to do this as I have highlighted below;

- Simplicity: *All you have to do is upload your web folders and files to a container.*
- Saving on hosting cost: *Cheaper compared to buying hosting fee from a web hosting company.*
- Better Performance: *Azure CDN can be configured to get a better performance delivering contents to your end users around the world.*

Combining Azure CDN with blob storage provides a low-cost and highly reliable static website hosting solution. These static sites have only CSS, HTML, JS files, fonts, etc.

## PREREQUISITES

There are some prerequisites we have to put in place before we proceed with this demo:

- Create a Microsoft Azure account, if you currently don't have one [here](#)
- Create a subscription
- Have a registered domain

Build in the cloud with an  
Azure free account

Create, deploy and manage applications across multiple clouds,  
on-premises and at the edge

Start free

Pay as you go >

Popular services free for 12 months

40+ other services free always

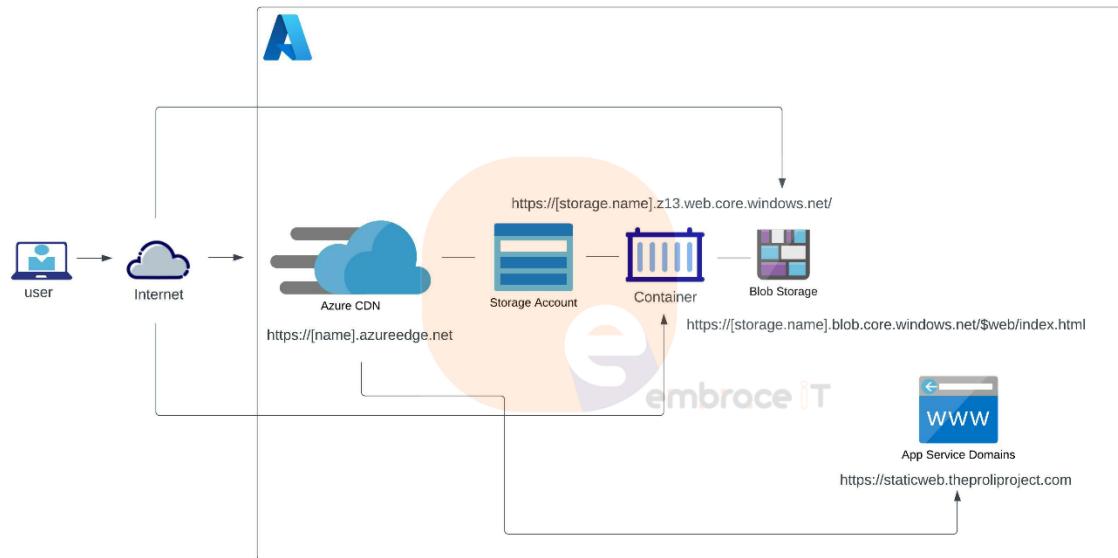
+ Start with USD200\* Azure credit

You will have 30 days to use it—in addition to free services.

**NOTE:** To those that are new to the Azure platform, when you create a free account, you get a \$200 free Azure credit to create resources for 30 days and a subscription is automatically created.

## OUTLINE OF WORKFLOW

Below is the workflow of what we would be doing in this demo,



Our static website would have a custom domain with which our end users would be able to access our website from the public web.

After all prerequisites has been put in place, let's get started.

- **CREATE A RESOURCE GROUP**

A resource group is a container where all related resources reside in.

Now let's proceed and create a resource group. See below images for visual guide.

The screenshot shows the Microsoft Azure portal homepage. At the top, there is a navigation bar with various links and a search bar. Below the navigation bar, the 'Azure services' section is visible, featuring icons for 'Create a resource', 'Resource groups', 'Container instances', 'Azure Active Directory', 'Storage accounts', 'Front Door and CDN profiles', 'DNS zones', 'Virtual machines', 'App Service Domains', and 'More services'. The 'Create a resource' button is highlighted with a red box and an arrow pointing up to the 'Resource groups' icon. The main content area displays a list of resources under the 'Resources' tab, with 'Recent' selected. The list includes items like 'APTS-Bootcamp-July-VM', 'pipelineteststorage1', 'MSDN Platforms Subscription - \$100', etc. Each item has a small icon, a name, a type, and a last viewed timestamp.

The screenshot shows the 'Resource groups' page in the Microsoft Azure portal. At the top, there is a navigation bar with various links and a search bar. Below the navigation bar, the 'Resource groups' section is visible, featuring a 'Create' button highlighted with a red box, a 'Manage view' dropdown, a 'Refresh' button, an 'Export to CSV' button, an 'Open query' button, and an 'Assign tags' button. There are also filters for 'Subscription equals all' and 'Location equals all'. The main content area displays a list of resource groups on the left and their details on the right. The details include 'Subscription' (e.g., 'my\_main\_subscription\_VSES (RN - 25th)', 'MSDN Platforms Subscription - \$100') and 'Location' (e.g., 'East US 2', 'East US'). Each resource group entry has a three-dot menu icon at the end.

**Create a resource group**

**Basics**

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

**Project details**

Subscription \* **MSDN Platforms Subscription - \$100**

Resource group \* **static-website-demo-rg**

**Resource details**

Region \* **(US) East US**

**Review + create**   **< Previous**   **Next : Tags >**

**Resource groups**

**Notifications**

More events in the activity log → Dismiss all ▾

**Resource group created**

Creating resource group 'static-website-demo-rg' in subscription 'MSDN Platforms Subscription - \$100' succeeded.

**Go to resource group**   **Pin to dashboard**

a few seconds ago

**\$72.31 credit remaining**

Subscription 'MSDN Platforms Subscription - \$100' has a remaining credit of \$72.31.

21 minutes ago

**\$96.91 credit remaining**

Subscription 'my\_main\_subscription\_VSES (RN - 25th)' has a remaining credit of \$96.91.

21 minutes ago

**Name**

- DefaultResourceGroup-EU2
- DefaultResourceGroup-WUS
- Devops
- NetworkWatcherRG
- practise
- proj\_project\_rg
- Repos
- Repos-eastus
- ResourceMoverRG-centralus-eastus-eus2
- static-website-demo-rg**
- t\_devops
- tfmaining
- VPNconnection\_group

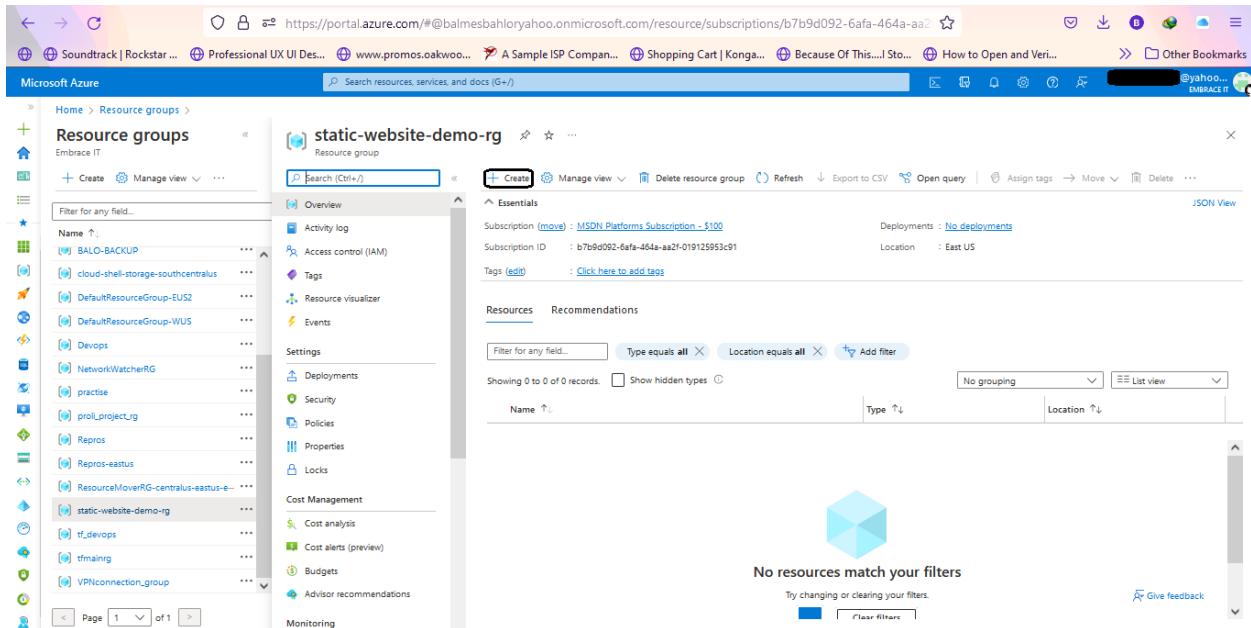
Subscription ↑↓

Subscription	Resource Group
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
MSDN Platforms Subscription - \$100	MSDN Platforms Subscription - \$100
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
MSDN Platforms Subscription - \$100	MSDN Platforms Subscription - \$100
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)
my_main_subscription_VSES (RN - 25th)	my_main_subscription_VSES (RN - 25th)

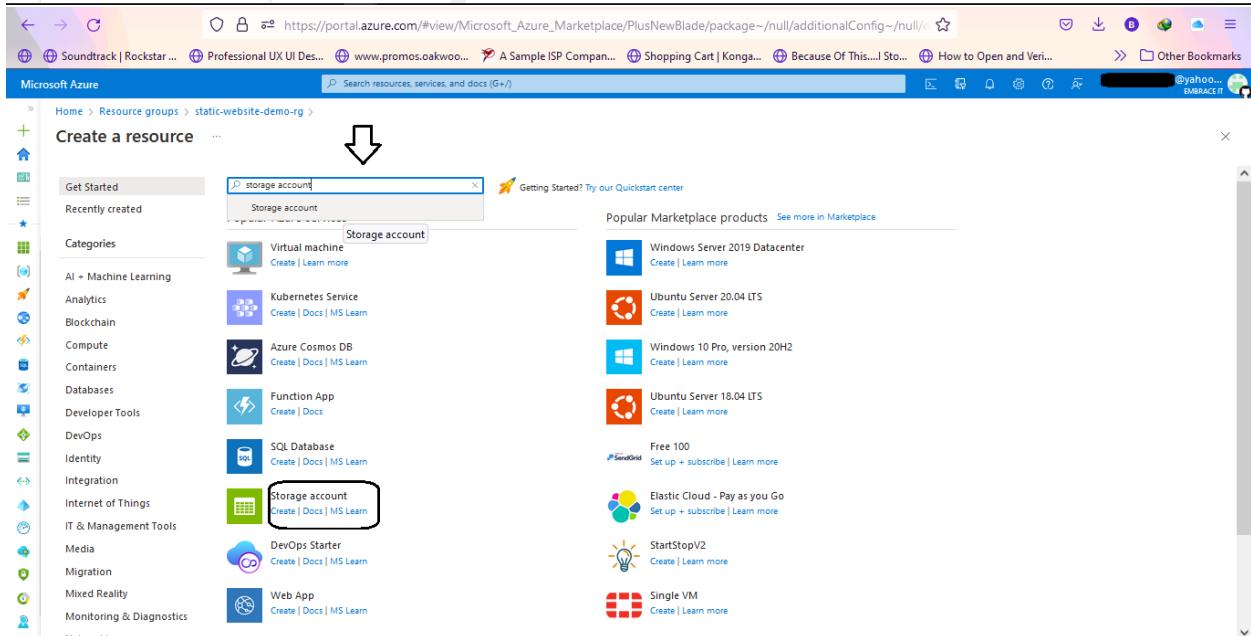
< Previous   Page 1 of 1   Next >   Showing 1 to 20 of 20 records.

## • CREATING A STORAGE ACCOUNT

After creating the resource group **static-website-demo-rg** for this demo, the next step is to create a storage account inside the **static-website-demo-rg** RG. See images below,



The screenshot shows the Microsoft Azure Resource Groups page. On the left, there's a sidebar with a list of resource groups, including 'static-website-demo-rg'. The main area shows the details for 'static-website-demo-rg' under the 'Overview' tab. It displays the subscription information ('Subscription (move) : MSDN Platforms Subscription - \$100'), subscription ID ('b7b9d092-6afa-464a-aaf2-01912353c01'), location ('East US'), and tags ('Click here to add tags'). Below this, the 'Resources' section shows a message: 'No resources match your filters' with a link to 'Try changing or clearing your filters.' and a 'Clear filters' button.

The screenshot shows the Microsoft Azure Marketplace search results for 'storage account'. The search bar at the top has 'storage account' typed into it. In the center, there's a list of products, with 'Storage account' highlighted by a red oval. To the right, there's a section titled 'Popular Marketplace products' listing various services like Windows Server 2019 Datacenter, Ubuntu Server 20.04 LTS, and Windows 10 Pro, version 20H2.

**Project details**

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription: MSDN Platforms Subscription - \$100

Resource group: static-website-demo-rg

**Instance details**

If you need to create a legacy storage account type, please click [here](#).

Storage account name: staticwebsitehosting1

Region: (US) East US

Performance: Standard: Recommended for most scenarios (general-purpose v2 account)

**Review + create**

**Validation passed**

**Basics**

Subscription	MSDN Platforms Subscription - \$100
Resource Group	static-website-demo-rg
Location	eastus
Storage account name	staticwebsitehosting1
Deployment model	Resource manager
Performance	Standard
Replication	Locally-redundant storage (LRS)

**Advanced**

Secure transfer	Enabled
Allow storage account key access	Enabled
Allow cross-tenant replication	Enabled
Default to Azure Active Directory authorization in the Azure portal	Disabled
Blob public access	Enabled

**Create**

Goto the newly created storage account resource which is **staticwebsitehosting1** for this demo. This is where most of the task would take place. As shown in image below, this is the interface you should see when you goto the storage account.

Click on capabilities and you see that the storage account at the moment is not configured for static website hosting and Azure CDN is also not configured as shown in the image above.

## **ENABLING STATIC WEBSITE ON STORAGE ACCOUNT**

Now click on static website and enable the storage account to host static content webpages.

Enter the **index.html** inside the index document block, you can also specify a root file for 404 error but we are not interested in that for this demo. Save and when it is enabled a primary endpoint

**https://staticwebsitehosting1.z13.web.core.windows.net** is created automatically this is just a generic URL for your end users to access your website contents.

See images below

The screenshot displays two consecutive screenshots of the Microsoft Azure Storage account settings for a static website.

**Screenshot 1:** The 'Static website' configuration page. The 'Static website' section has the 'Enabled' button selected. Other settings include 'Index document name' set to 'index.html' and 'Error document path' left blank. A status message at the top states: "Enabling static websites on the blob service allows you to host static content. Webpages may include static content and client-side scripts. Server-side scripting is not supported. As data is replicated asynchronously from primary to secondary regions, files at the secondary endpoint may not be immediately available or in sync with files at the primary endpoint." A progress bar indicates "Updating static website settings..."

**Screenshot 2:** The same configuration page after saving changes. The 'Enabled' button remains selected. A message at the top says: "An Azure Storage container has been created to host your static website." A red arrow points to the 'Save' button in the top right corner of the form.

Also, after enabling static website a container **\$web** is automatically created, this container would hold your web content files and folders.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is https://portal.azure.com/#@balmesbahlyahoo.onmicrosoft.com/resource/subscriptions/b7b9d092-6afa-464a-aa... . The main content area displays the 'Containers' section for the storage account 'staticwebsitehosting1'. A search bar at the top right contains the placeholder 'Search resources, services, and docs (G+)'. Below it, a toolbar includes icons for Container, Change access level, Restore containers, Refresh, and Delete. A table lists the containers: '\$logs' (Last modified: 7/9/2022, 7:43:28 PM, Public access level: Private, Lease state: Available) and '\$web' (Last modified: 7/9/2022, 7:47:22 PM, Public access level: Private, Lease state: Available). On the left, a sidebar navigation menu includes sections like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser (preview), Data storage (Containers, File shares, Queues, Tables), Security + networking (Networking, Azure CDN, Access keys, Shared access signature), and Execution. An upward arrow icon is overlaid on the left side of the container list.

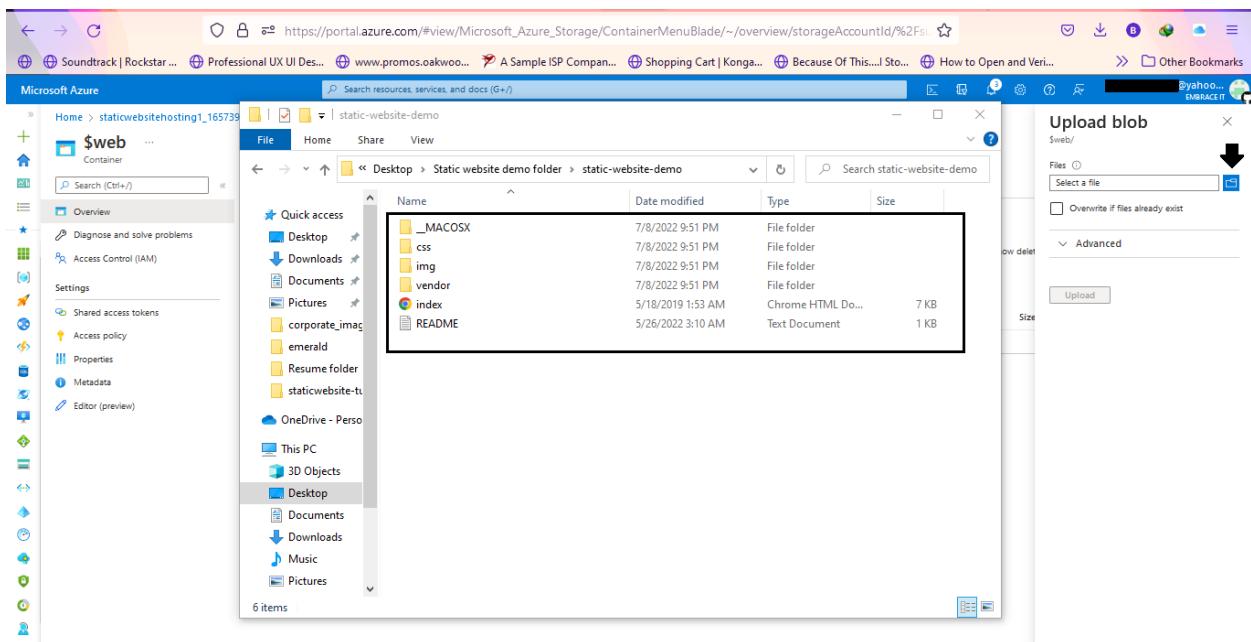
If we check capabilities under the storage account overview, we see that static website has been configured.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is https://portal.azure.com/#@balmesbahlyahoo.onmicrosoft.com/resource/subscriptions/b7b9d092-6afa-464a-aa... . The main content area displays the 'Overview' section for the storage account 'staticwebsitehosting1'. At the top, there are buttons for Upload, Open in Explorer, Delete, Move, Refresh, Mobile, and Feedback. A 'JSON View' link is located in the top right. Below these are sections for Resource group, Location, Subscription, and Disk state. Under the 'Properties' tab, there are tabs for Properties, Monitoring, Capabilities (7), Recommendations, Tutorials, and Developer Tools. The 'Capabilities' tab is currently selected. It shows several configuration status boxes: 'Static website' (Host static content on the blob service, status: Configured), 'Data protection' (Save and recover data when it is erroneously modified or deleted, status: Failed to load configuration status), 'Lifecycle management' (Create rule-based policies for storage accounts, status: Not configured), 'Custom domain' (Configure a custom domain for accessing blob data, status: Not configured), 'Security' (Enable Azure Defender for your storage account, status: Failed to load configuration status), and 'Private endpoints' (Secure data access over a private link, status: Not configured). An upward arrow icon is overlaid on the left side of the container list.

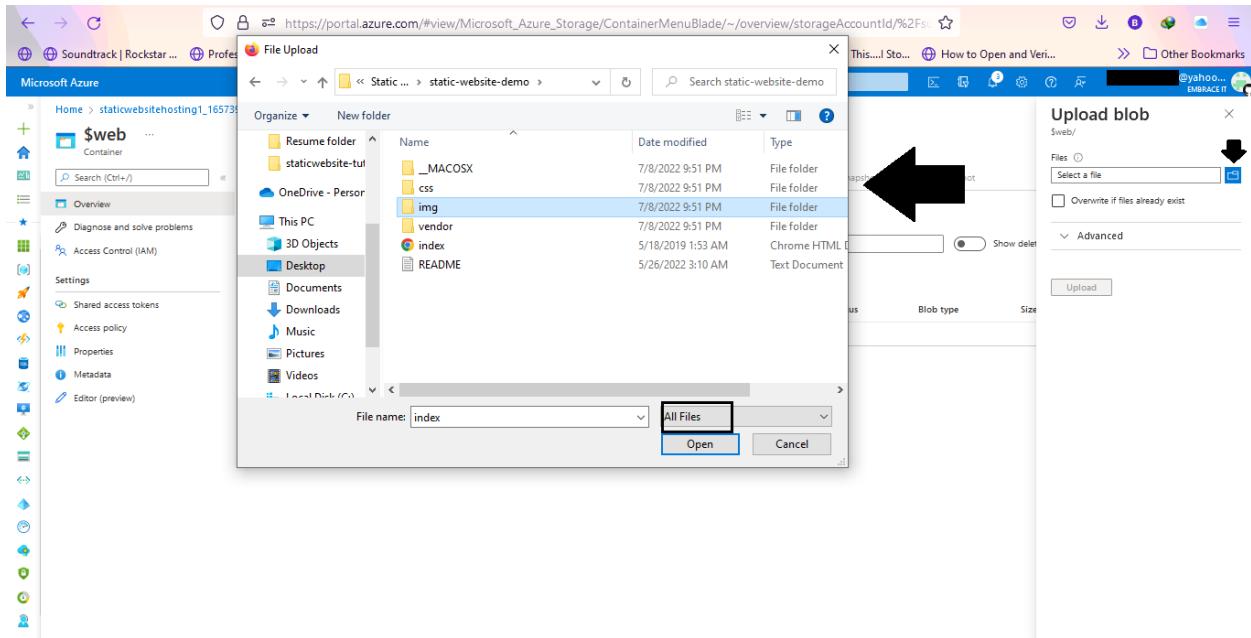
## UPLOADING WEB CONTENTS TO CONTAINER

Now it's time to upload our website contents to the container **\$web**. You might be thinking uploading the web content to the container from the portal should be easy by just clicking the upload button and upload the web folders and files to the container. Unfortunately, you cannot upload folders to the container from the portal directly except for files.

Here is our web content below,



see below image you can only upload files to the container. For our web contents only *index.html* and *README.txt* can be uploaded which does not work for us because all folders and files has to be uploaded

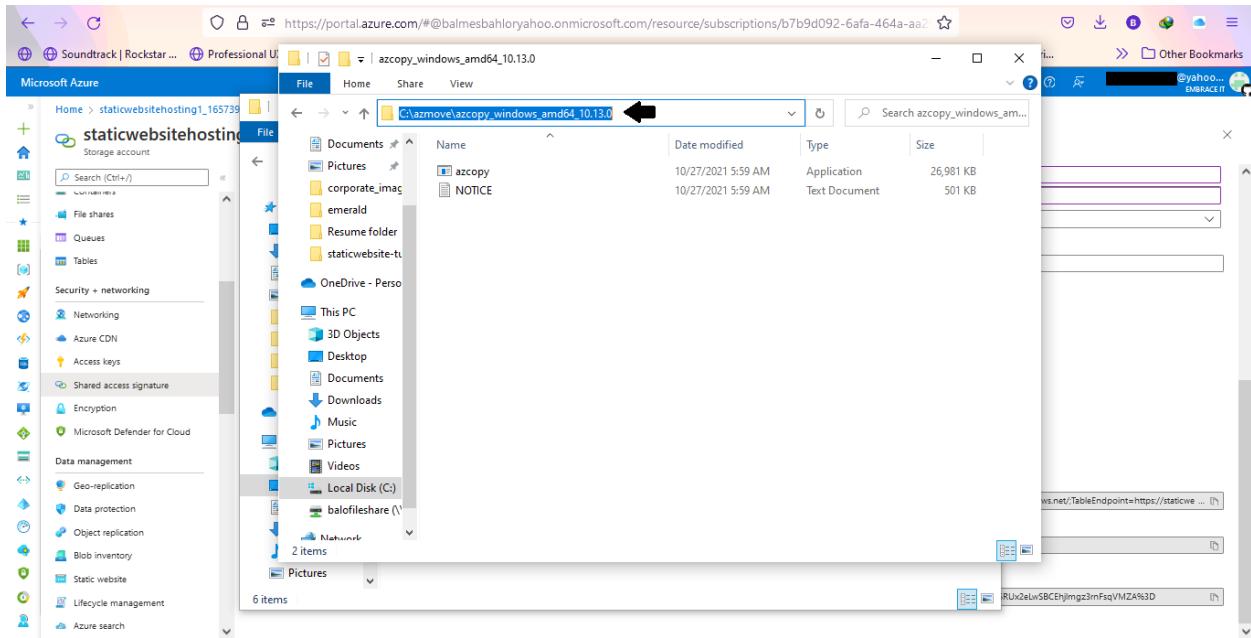


- **DOWNLOADING AZCOPY**

To upload our web contents to the container we would make use a tool called AzCopy. AzCopy is a CLI (command-line) tool that moves data into and out of an Azure Storage.

Next step is to download the AzCopy tool that best fits your environment here -> <https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10#download-azcopy>.

Unzip the AzCopy file, for this demo I have the AzCopy unzipped folder copied to a folder "azmove" in my C:\ drive root. You can add the directory location of the AzCopy executable to your system environment variable path for ease of use, but for this I will navigate to the folder where we have the AzCopy executable and copy the file full path somewhere like a notepad. See Image below



- **AUTHORIZE AZCOPY TO ACCESS STORAGE ACCOUNT**

We would need to authorize AzCopy to access the storage account so we can upload the web contents to the container. We would provide authorization credentials to AzCopy by using **shared access signature (SAS) token** generated from the storage account. Click on the shared access signature on the left pane under "security + networking".

Explaining the key options

**Allowed service** - check *blob* only because we want to take into consideration one of cloud best practices which is least privilege concept. This SAS token is only intended for the blob service and no access should be granted to your file, queue or table service using the generated SAS token.

**Allowed resource type** - check container as this is the only resource, we want the SAS token to access.

**Start and expiry date/time** - set the period you want this SAS token to be valid through. Put into consideration security best practices, expiry date shouldn't elapse the period SAS token is expected to be used.

Leave other options as it is and proceed to click on generate SAS and connection string.

See Images below,

**staticwebsitehosting1 | Shared access signature**

**Containers**

Allowed services: Blob, Container, Object

Allowed resource types: Service, Container, Object

Allowed permissions: Read, Write, Delete, List, Add, Create, Update, Process, Immutable storage

Blob versioning permissions: Enables deletion of versions

Allowed blob index permissions: Read/Write, Filter

Start and expiry date/time:

- Start: 07/09/2022
- End: 07/10/2022

(UTC+01:00) West Central Africa

Allowed IP addresses: For example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols: HTTPS only

Preferred routing tier: Basic (default)

Signing key: key1

**Generate SAS and connection string**

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons. The main area is titled 'staticwebsitehosting1 | Shared access signature'. It has sections for 'Start' (07/09/2022) and 'End' (07/10/2022), 'Allowed IP addresses' (empty), 'Allowed protocols' (HTTPS only selected), 'Preferred routing tier' (Basic selected), and a 'Signing key' dropdown set to 'key1'. Below these are fields for 'Connection string' and 'SAS token'. The 'SAS token' field contains a long URL starting with '?sv=2021-06-08&ss=b&srt=c&sp=rw&lacitfx&se=2022-07-10T03:24:20Z&st=2022-07-09T19:20Z&spr=https&sig=' followed by a long hex string.

SAS token is generated as you can see in the last image above.

Copy the SAS token somewhere like a notepad, once you leave this page you cannot retrieve the token any longer. As you can see in the image below I have copied all important path, urls, token to one point.

The screenshot shows a Windows Notepad window with the title '\*Untitled - Notepad'. It contains the following text:

```

.\azcopy copy "[path to the website folder on your local pc]" "{path to the
container hosting the website}{SAS token}" --recursive
.\azcopy copy
C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo
https://staticwebsitehosting1.blob.core.windows.net/$web
?sv=2021-06-08&ss=b&srt=c&sp=rw&lacitfx&se=2022-07-10T03:24:20Z&st=2022-07-
09T19:20Z&spr=https&sig=5o88GgM9Qu980YjpB1Oyb2AozXKDLPNNI7YA%2FeDzSdc%3D

```

A black arrow points from the text to the SAS token in the third line. To the right of the notepad, a context menu is open with the option 'Copied' highlighted.

Now let me bring your attention to the AzCopy commands we would run to upload our web contents to the container.

## AzCopy script

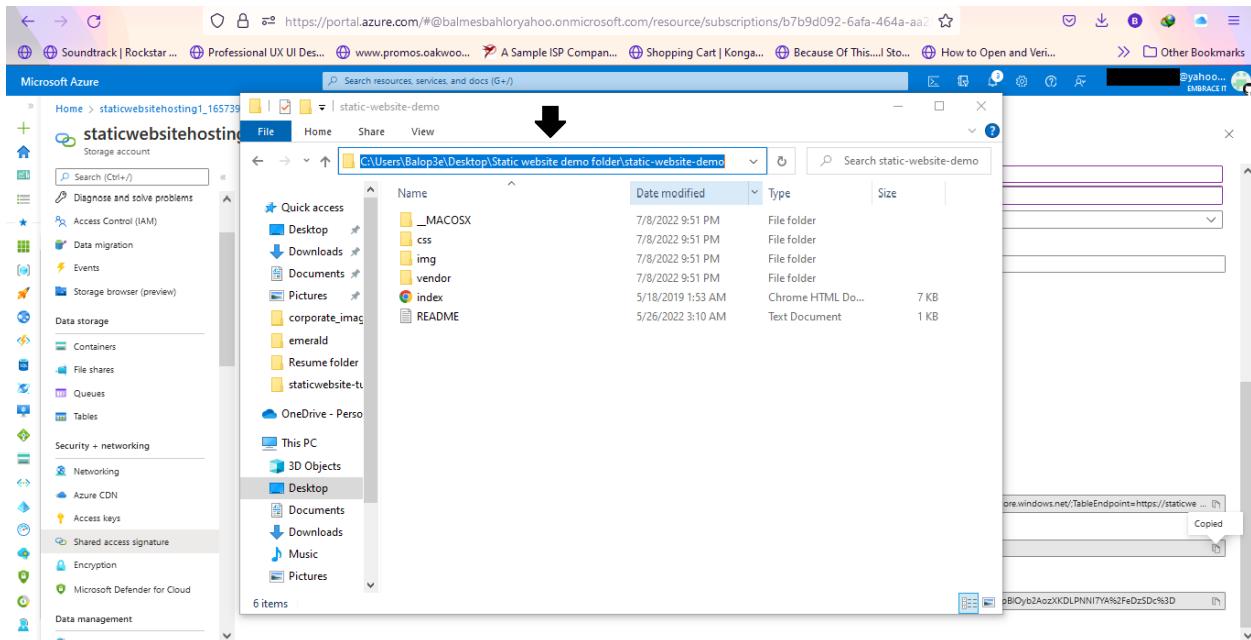
```
.\azcopy copy "{path to the website folder on your local pc}" "{path to the container hosting the website}{SAS token}" --recursive
```

### Explaining the contents of AzCopy scripts

**.\azcopy copy** - This command copies source data to a destination location, be it from your local computer to a storage account or from a storage account to your local computer.

**--recursive** - This flag is necessary because we want to make sure it copies all sub-folders inside folders and files

**{path to the website folder on your local pc}** - This is the source destination of your web contents, to be uploaded to the container. See image below



**{path to the container hosting the website}** - This is the target destination which is your container. To get the URL/path to your container see the images below

The image consists of two screenshots of the Microsoft Azure portal. The top screenshot shows the 'Containers' blade for a storage account named 'staticwebsitehosting1'. It lists two containers: 'Slogs' and '\$web'. A context menu is open over the '\$web' container, with the 'Container properties' option highlighted. The bottom screenshot shows the 'Properties' blade for the '\$web' container. The URL field contains the value 'https://staticwebsitehosting1.blob.core.windows.net/\$web'. A tooltip 'Copied' is visible next to a copy icon.

Now if you remember we are making use of AzCopy because we cannot upload folders to the container using Azure portal, if we copy the source folder to the container, our web content would be nested inside another folder and this would not work. We need all folder and files of the web content to sit at the root of the container. I have created an AzCopy scripts for each folder and file to be run and pushed to the container. see image below

```

Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

+ C:\Users\Balop3e>cd C:\azmove\azcopy_windows_amd64_10.13.

C:\azmove\azcopy_windows_amd64_10.13.0

*Untitled - Notepad
File Edit Format View Help
.\azcopy copy "{path to the website folder on your local pc}" "{path to the container hosting the website}{SAS token}" --recursive

.\azcopy copy "C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo\css" "https://staticwebsitehosting1.blob.core.windows.net$web?sv=2021-06-08&ss=b&srt=sc&p=rwdlacitfx&se=2022-07-10T03:36:31Z&st=2022-07-09T19:36:31Z&spr=https&sig=rBF5Z8Wyz51QdMdJicTdX2Bw2%2FIL4MfhFvNIU%2FhUz0LQo%3D" --recursive

.\azcopy copy "C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo\MACOSX" "https://staticwebsitehosting1.blob.core.windows.net$web?sv=2021-06-08&ss=b&srt=sc&p=rwdlacitfx&se=2022-07-10T03:36:31Z&st=2022-07-09T19:36:31Z&spr=https&sig=rBF5Z8Wyz51QdMdJicTdX2Bw2%2FIL4MfhFvNIU%2FhUz0LQo%3D" --recursive

.\azcopy copy "C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo\img" "https://staticwebsitehosting1.blob.core.windows.net$web?sv=2021-06-08&ss=b&srt=sc&p=rwdlacitfx&se=2022-07-10T03:36:31Z&st=2022-07-09T19:36:31Z&spr=https&sig=rBF5Z8Wyz51QdMdJicTdX2Bw2%2FIL4MfhFvNIU%2FhUz0LQo%3D" --recursive

.\azcopy copy "C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo\vendor" "https://staticwebsitehosting1.blob.core.windows.net$web?sv=2021-06-08&ss=b&srt=sc&p=rwdlacitfx&se=2022-07-10T03:36:31Z&st=2022-07-09T19:36:31Z&spr=https&sig=rBF5Z8Wyz51QdMdJicTdX2Bw2%2FIL4MfhFvNIU%2FhUz0LQo%3D" --recursive

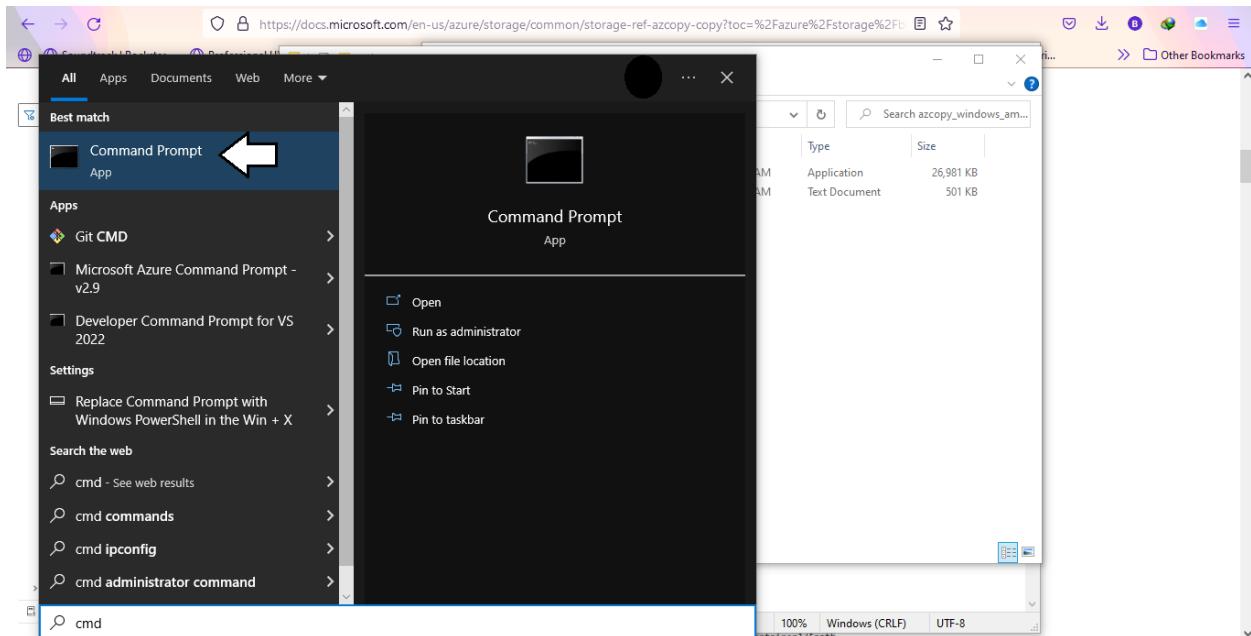
.\azcopy copy "C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo\index.html" "https://staticwebsitehosting1.blob.core.windows.net$web?sv=2021-06-08&ss=b&srt=sc&p=rwdlacitfx&se=2022-07-10T03:36:31Z&st=2022-07-09T19:36:31Z&spr=https&sig=rBF5Z8Wyz51QdMdJicTdX2Bw2%2FIL4MfhFvNIU%2FhUz0LQo%3D" --recursive

.\azcopy copy "C:\Users\Balop3e\Desktop\Static website demo folder\static-website-demo\README.txt" "https://staticwebsitehosting1.blob.core.windows.net$web?sv=2021-06-08&ss=b&srt=sc&p=rwdlacitfx&se=2022-07-10T03:36:31Z&st=2022-07-09T19:36:31Z&spr=https&sig=rBF5Z8Wyz51QdMdJicTdX2Bw2%2FIL4MfhFvNIU%2FhUz0LQo%3D" --recursive

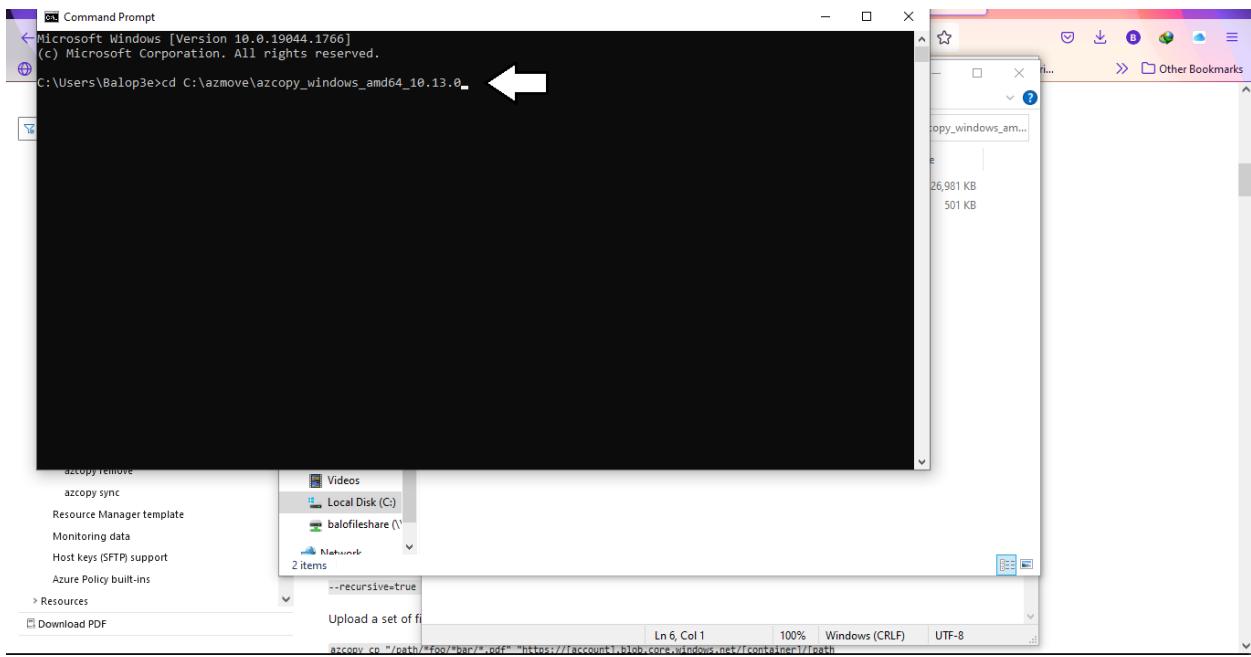
```

## • RUNNING THE AZCOPY SCRIPT

On your local pc launch **cmd** or **PowerShell** to run the AzCopy script. For this demo I am making use of cmd.

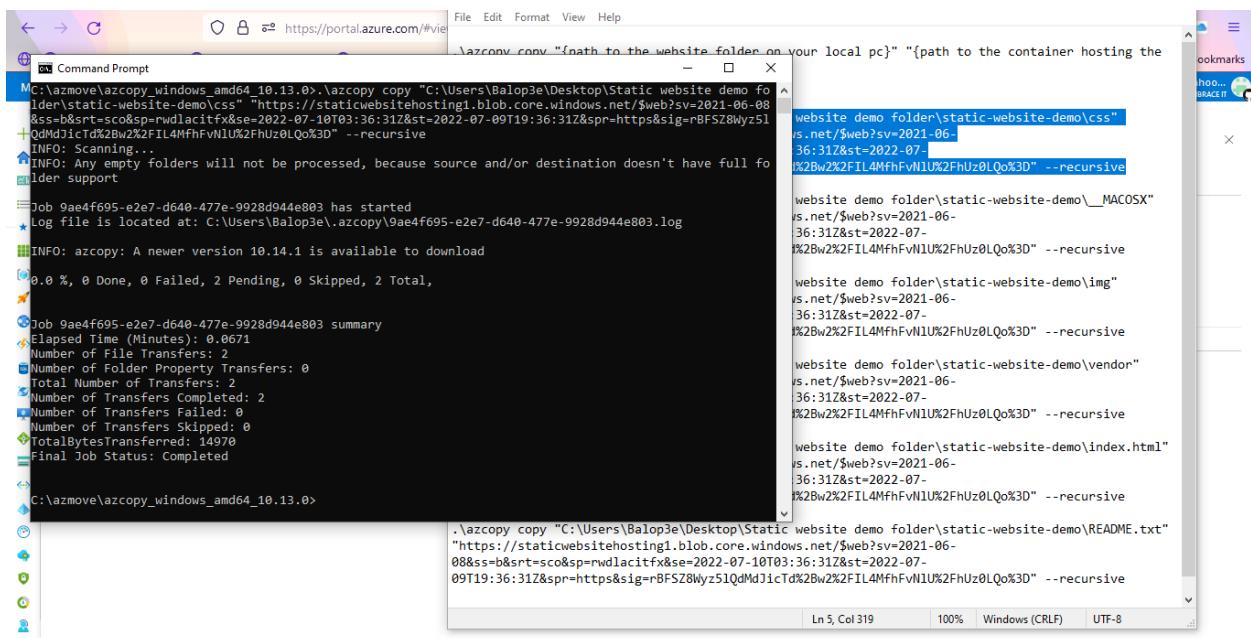


Change the current working directory to where you have the AzCopy executable file by running "**cd {file path to where the azcopy exec file is}**" as shown below



Copy the AzCopy script we coined together on notepad and past the script in cmd and run one after the other.

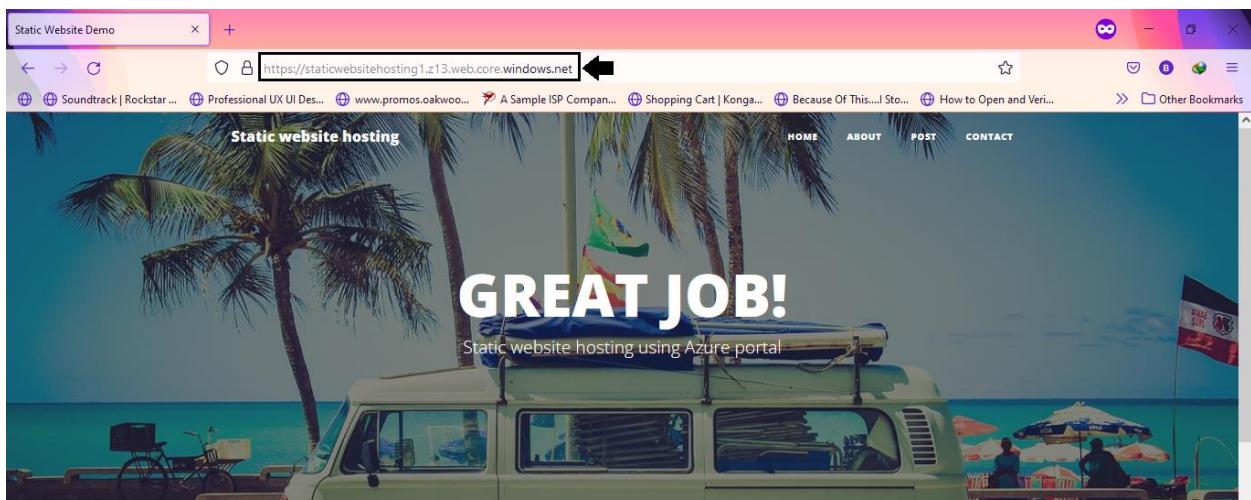
**Note:** Do this for all the remaining scripts.



After successfully uploading the web contents, and you refresh your container you would get your web contents inside the **\$web** container.

The screenshot shows the Microsoft Azure Storage Container blade for the '\$web' container. On the left, there's a sidebar with various options like Overview, Diagnose and solve problems, Access Control (IAM), and Settings. The main area shows a table of blobs with columns for Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. The table includes entries for 'index.html' (Modified 7/9/2022, 9:03:29 PM, Hot (Inferred), Block blob, 6.14 KB, Available) and 'README.txt' (Modified 7/9/2022, 9:03:43 PM, Hot (Inferred), Block blob, 382 B, Available).

Remember the primary endpoint link <https://staticwebsitehosting1.z13.web.core.windows.net/> that was generated when you enabled static website on your storage account? now that is the link for your end users to access your content on the public web. See image below



### **Next Demo:- Static website hosting using Azure DevOps**

The next demo we would be hosting a static website using Github, terraform and Azure DevOps pipelines.

## CONFIGURING AZURE CDN

Now it's time to configure Azure CDN, firstly let's try to understand what Azure CDN is and its benefits to our end users.

Azure Content Delivery Network (CDN) is designed to send web contents such as audio, video, images, and other files faster and more reliably to your end users using servers that are closest to the end users. This helps to increase speed and availability, resulting in significant user experience improvements.

Below image shows Azure CDN is not configured yet.

The screenshot shows the Azure Storage account configuration page for 'staticwebsitehosting1'. The 'Capabilities' tab is selected, displaying several service configurations:

- Static website: Host static content on the blob service. Status: Configured
- Data protection: Save and recover data when it is erroneously modified or deleted. Status: Partially configured
- Lifecycle management: Create rule-based policies for storage accounts. Status: Not configured
- Azure CDN: Serve media files quickly and reliably with the Azure CDN. Status: Not configured (highlighted with a red arrow)
- Custom domain: Configure a custom domain for accessing blob data. Status: Not configured
- Security: Enable Azure Defender for your storage account. Status: Not configured
- Private endpoints: Secure data access over a private link. Status: Not configured

Click on Azure CDN on the left pane under the “*security + networking*” and create a new endpoint.

Give it a CDN profile name, select the pricing tier “*select Microsoft CDN (classic)*” give the CDN endpoint a name.

Origin hostname - you have to select the origin server from which CDN endpoint pulls content from. I would be using my primary endpoint which is “**staticwebsitehosting1.z13.web.core.windows.net**” now proceed to create the CDN. See image below,

The screenshot shows two consecutive steps in the Azure portal for creating a new CDN endpoint.

**Step 1: Initial Endpoint Creation**

The 'Endpoints' section is displayed. A table header is shown with columns: Hostname, Status, Protocol. Below it, a note says 'Create new CDN endpoints below.' and 'Loading'. At the bottom, there's a 'Migrate custom domains to CDN' link and a 'New endpoint' section. In the 'New endpoint' section, the 'CDN profile' dropdown is set to 'Create new' (highlighted with a red arrow), and the 'Create' button is also highlighted with a red arrow.

**Step 2: Detailed Endpoint Configuration**

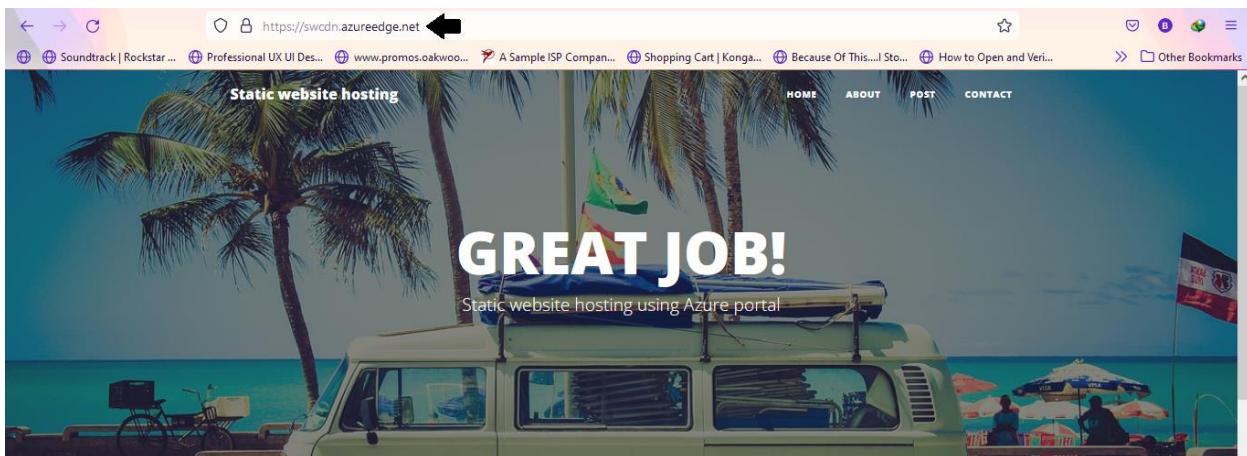
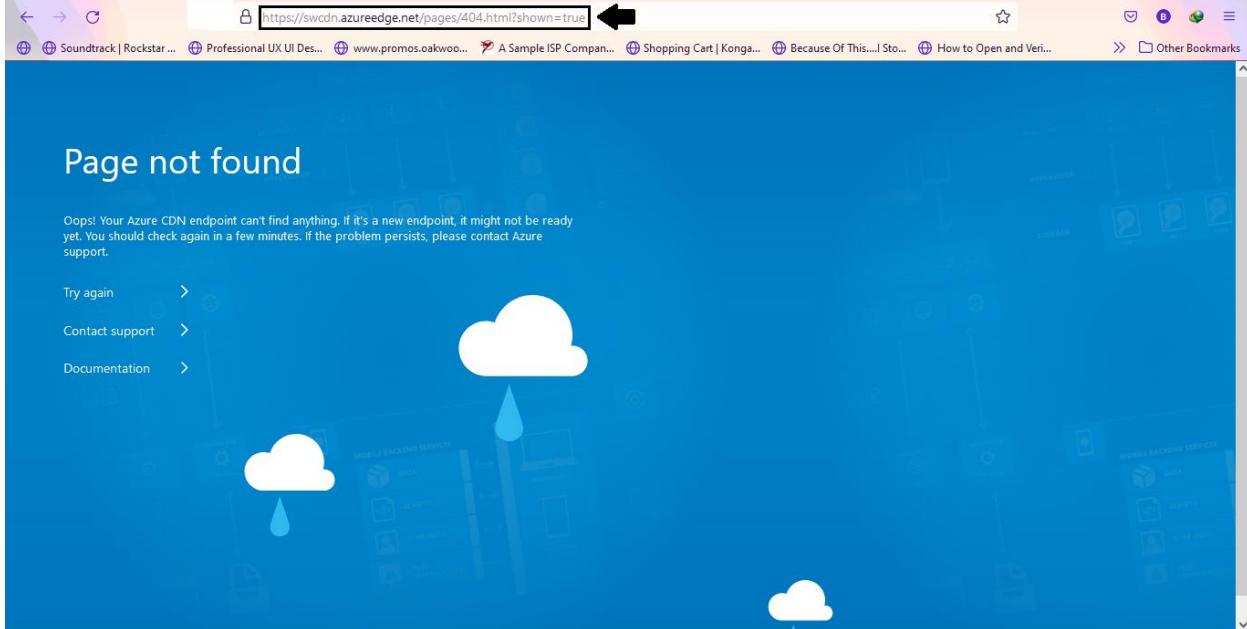
The 'Endpoints' section is shown again. The 'New endpoint' configuration is detailed. The 'CDN profile' dropdown is set to 'Create new' (highlighted with a red arrow). The 'Pricing tier' dropdown is set to 'Microsoft CDN (classic)' (highlighted with a red arrow). The 'CDN endpoint name' input field contains 'svcdn' (highlighted with a red arrow). The 'Origin hostname' dropdown lists 'staticwebsitehosting1.z13.web.core.windows.net (Static website)', 'staticwebsitehosting1.blob.core.windows.net (Blob)', and 'staticwebsitehosting1.z13.web.core.windows.net (Static website)'. The 'Create' button at the bottom is highlighted with a red arrow.

The last image above shows our CDN endpoint is ready and running.

Click on the newly created CDN endpoint, you can see 2 URLs one is the CDN endpoint and the other is the origin hostname aka primary endpoint for our static website. See image below,

Now that Azure CDN has been successfully configured for our setup, we can proceed to load the URL of the CDN endpoint which is <https://swcdn.azureedge.net>

**Note:** it would take some time for Azure CDN to load your content you might get the blue screen error or "*the account being accessed does not support http*" error. Just give it sometime and refresh after some minutes. See Images below



### Next Demo:- Static website hosting using Azure DevOps

The next demo we would be hosting a static website using Github, terraform and Azure DevOps pipelines.

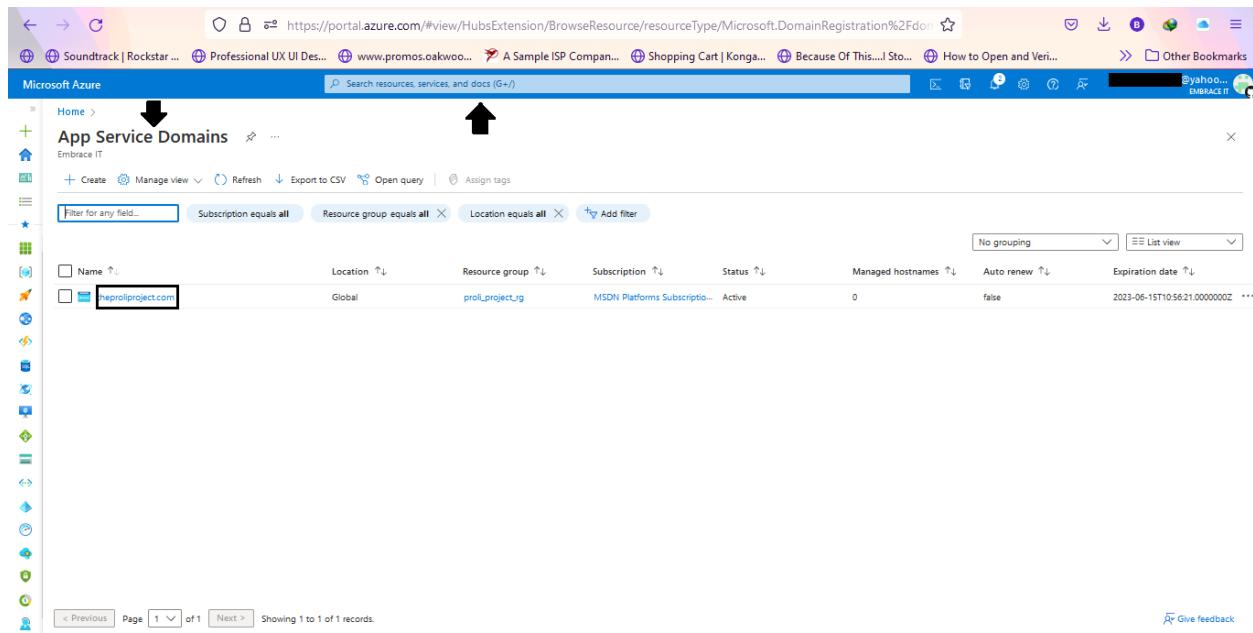
## CONFIGURING A CUSTOM DOMAIN TO YOUR AZURE CDN ENDPOINT

Now can you imagine your end users accessing your webpage using the URL <https://swcdn.azureedge.net> of course you wouldn't want that, you would want them to access your webpage using your preferred domain name which can be an existing one for your business or a newly registered domain name.

For this demo I would be using an existing domain I registered on Azure "[thepronliproject.com](http://thepronliproject.com)" and another registered on Namecheap "[mast\\*\\*\\*.org](http://mast***.org)". I will be making 2 demonstrations for domain registered on Azure portal and domain registered by a different registrar.

- Domain registered on Azure portal.

Goto App Service Domains and select the domain. See image below



Name	Location	Resource group	Subscription	Status	Managed hostnames	Auto renew	Expiration date
thepronliproject.com	Global	pronliproject_rg	MSDN Platforms Subscription	Active	0	false	2023-05-15T10:56:21.000000Z

What we want to do? we want to create a CNAME record with the DNS provider for "[thepronliproject.com](http://thepronliproject.com)" that points to "[swcdn.azureedge.net](https://swcdn.azureedge.net)" doing this we would be able to associate the domain to the CDN endpoint. After selecting the domain, click on *manage DNS records* and click on *record set*.

For this demo I am making use of a sub domain **staticweb.theproliproject.com**

- Enter the name i.e. the subdomain name
- Enter the type - you are creating a CNAME record, this is because you are mapping/pointing your host name to another name record which is "*swcdn.azureedge.net*"
- Leave Alias record set to No
- TTL - leave at default
- Alias - is the CDN endpoint "*swcdn.azureedge.net*"

Click ok after entering the appropriate information's. See images below

<https://portal.azure.com/#@balmesbahli@yahoo.onmicrosoft.com/resource/subscriptions/b7b9d092-6afa-464a-aa2f-019125953c91/resourceGroups/prol-project-rg/providers/Microsoft.AppService/domains/theproliproject.com>

<https://portal.azure.com/#@balmesbahli@yahoo.onmicrosoft.com/resource/subscriptions/b7b9d092-6afa-464a-aa2f-019125953c91/resourceGroups/prol-project-rg/providers/Microsoft.Dns/dnszones/theproliproject.com>

**App Service Domains**

**theproniproject.com**

**Essentials**

Resource group (move) : <a href="#">prol-project-rg</a>	Domain : <a href="#">theproliproject.com</a>
Location : Global	Expiration date : 2023-06-15T105621
Subscription (move) : <a href="#">MSDN Platforms Subscription - \$100</a>	Auto renew : Disabled
Subscription ID : b7b9d092-6afa-464a-aa2f-019125953c91	Privacy protection : Enabled
Status : Active	
Tags (edit) : prol_project:domain-name	

**DNS zone**

**Record set**

Name	Type	TTL	Value	Alias resource type	Alias target
@	NS	172800	ns1-03.azure-dns.com. ns2-03.azure-dns.net. ns3-03.azure-dns.org. ns4-03.azure-dns.info.		...
@	SOA	3600	Email: azuredns-hostmaster.micr... Host: ns1-03.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2199200 Minimum TTL: 300 Serial number: 1		...
@	MX	3600	0 theproniproject.com.mail.protoc...		...
@	TXT	3600	v=spf1 includespf.protection.out...		...
autodiscover	CNAME	3600	autodiscover.outlook.com		...
mail	TXT	3600	v=spf1 includespf.protection.out...		...

The screenshot shows the Microsoft Azure portal interface for managing DNS zones. On the left, the navigation pane includes links for Home, App Service Domains, theproliproject.com, DNS zone, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Properties, Locks, Monitoring, Alerts, Metrics, Automation, Tasks (preview), Export template, Support + troubleshooting, Resource health, New Support Request), and Help & feedback. The main content area displays the DNS zone for 'theproliproject.com'. A search bar at the top right says 'Search resources, services, and docs (G+)'. The DNS zone page lists several record sets, including NS, SOA, MX, TXT, CNAME, and a newly created CNAME record for 'staticweb'. A modal window titled 'Add record set' is open on the right, showing the configuration for the new CNAME record. The 'Name' field is set to 'staticweb', the 'Type' is 'CNAME - Link your subdomain to another record', and the 'Alias' field is set to 'swcdn.azureedge.net'. The 'OK' button is highlighted with a red arrow. Below the modal, a notification bar shows a success message: 'Successfully created record set 'staticweb''. Other notifications include '\$96.13 credit remaining' and '\$72.22 credit remaining'.

After successfully creating the DNS record, we can proceed to Azure CDN and map our custom domain to Azure CDN. Goto the Azure CDN and click on custom domain

Enter custom hostname - enter your subdomain here for this demo  
**"staticweb.theproliproject.com"** click add

The screenshot illustrates the steps to add a custom domain to an Azure CDN endpoint. In the first part, the 'Custom domains' blade is shown with the 'Add' button highlighted. In the second part, the 'Notifications' blade shows several successful creation messages, indicating that the custom domain has been successfully added.

If you load the **staticweb.theproliproject.com** in your web browser you would get the **warning: potential security risk ahead** warning message. If you proceed to accept the risk and continue the website loads but with the SSL certificate missing as shown in image below



### **Next Demo:- Static website hosting using Azure DevOps**

The next demo we would be hosting a static website using Github, terraform and Azure DevOps pipelines.

Now let us proceed to enable https which enables you to deliver your web contents to your end users securely over the public web.

To enable https, click on the custom domain just created and toggle the enable https feature.

See image below

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is [https://portal.azure.com/#view/Microsoft\\_Azure\\_Cdn/CustomDomainHttpsBlade/customDomainId/%2Fsubscription%2F](https://portal.azure.com/#view/Microsoft_Azure_Cdn/CustomDomainHttpsBlade/customDomainId/%2Fsubscription%2F). The page displays a 'Custom domain' configuration for the domain [staticweb.theproliproject.com](https://staticweb.theproliproject.com). A black arrow points to a toggle switch labeled 'Custom domain HTTPS' with the 'On' option selected. The left sidebar contains various Azure service icons.

After enabling https, Azure then undergoes the following processes;

- Submits your https request
- Validates your domain.
- Provision a certificate for your domain.

See image below,

**About This Feature**

Custom Domain HTTPS feature enables you to deliver content to your users securely over your own domain. This is done by encrypting the traffic between the CDN and your user's client (typically web browsers) via TLS protocol (which is a standard for SSL protection used by most websites). Using this feature, you can enable HTTPS on your custom domain with just a few clicks and have Azure CDN completely take care of certificate management tasks such as its renewal. You can also bring your own certificate (stored in [Azure Key vault](#)) or even purchase a new certificate through Key vault and have Azure CDN use that certificate for securing the content delivery.

[Learn more](#)

**Configure**

Custom domain HTTPS:  On  Off

Certificate management type:  CDN managed  Use my own certificate

Minimum TLS version:  TLS 1.2  TLS 1.0/1.1

**Status**

- 1 Submitting request  
Your HTTPS request is being submitted.
- 2 Domain validation  
Not started.
- 3 Certificate provisioning  
Not started.
- 4 Complete  
Not started.

**Notifications**

More events in the activity log → Dismiss all ▾

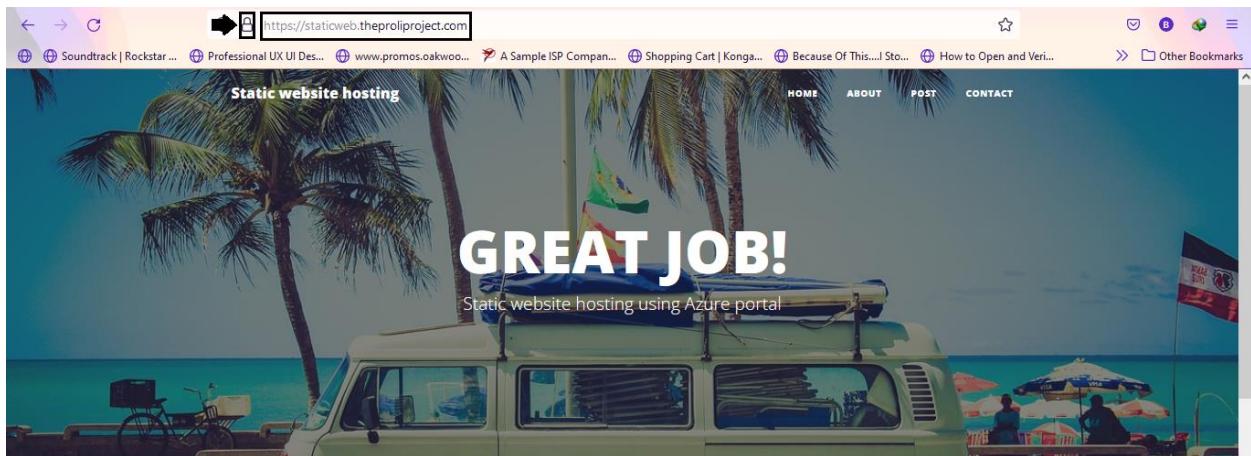
- Custom domain was successfully updated. Your request is being processed. This custom domain will be automatically validated if no CNAME record exists mapping this custom domain to the CDN endpoint. If no CNAME record exists, a validation email will be sent to the following email addresses. Please verify the domain as soon as possible. admin@your-domain-name.com, administrator@your-domain-name.com, webmaster@your-domain-name.com, hostmaster@your-domain-name.com, postmaster@your-domain-name.com a few seconds ago
- Successfully created a CDN custom domain It can take up to 10 minutes for staticweb.theproliproject.com settings to reach every CDN POP. 23 minutes ago
- Successfully created CDN endpoint Your CDN endpoint 'swcdn' was successfully created. 41 minutes ago
- Successfully created storage container Successfully created storage container 'sweb'. an hour ago
- Successfully updated static website settings Successfully updated static website settings for 'staticwebsitehosting1'. Settings may take up to 30 seconds to take effect. 2 hours ago

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is [https://portal.azure.com/#view/Microsoft\\_Azure\\_Cdn/CustomDomainHttpsBlade/customDomainId%2Fsubscription](https://portal.azure.com/#view/Microsoft_Azure_Cdn/CustomDomainHttpsBlade/customDomainId%2Fsubscription). The page title is "Custom domain". Under "Custom domain HTTPS", the "On" switch is turned on. The "Certificate management type" is set to "CDN managed". The "Minimum TLS version" is set to "TLS 1.2". A large box labeled "Status" contains four steps: 1. Submitting request (status: Your HTTPS request has been submitted successfully, with a green checkmark). 2. Domain validation (status: Your request is being processed. This custom domain will be automatically validated if a CNAME record exists matching this custom domain in the CDN endpoint. If no CNAME record exists, a validation email will be sent to the following email addresses: Please verify my domain as soon as possible. <admin@your-domain-name.com, administrator@your-domain-name.com, webmaster@your-domain-name.com, hostmaster@your-domain-name.com, postmaster@your-domain-name.com>, with a blue circular progress bar). 3. Certificate provisioning (status: Not started). 4. Complete (status: Not started).

After the process is completed, you can reload the URL in your browser and your website should be secured with a SSL certificate.

**Note:** it might take up to 5 minutes for this process to get completed.

See image below



### Next Demo:- Static website hosting using Azure DevOps

The next demo we would be hosting a static website using Github, terraform and Azure DevOps pipelines.

- Domain registered with another registrar

Login to your account dashboard of your registrar/host account and navigate to where you can manage your domain.

Enter a CNAME record with the host name which is your preferred subdomain name e.g., for this demo "staticweb" and the value which is the Azure CDN endpoint "swcdn.azureedge.net". See image below,

The screenshots illustrate the process of managing a domain registered with Namecheap. In the first screenshot, the user is on the main dashboard, navigating through the 'Domain List'. They find the domain they want to manage and click the 'MANAGE' button. In the second screenshot, they are on the detailed domain management page for 'mast[REDACTED].org'. They have selected the 'Advanced DNS' tab, which is highlighted with a large black box and an arrow pointing to it.

The screenshot shows the Namecheap Domain Control Panel interface. The left sidebar includes links for Dashboard, Expiring / Expired, Domain List (which is selected), Hosting List, Private Email, SSL Certificates, Apps, and Profile. The main content area has tabs for DNS Templates, Host Records, DNSSEC, and Mail Settings. Under Host Records, there is a table with columns Type, Host, Value, and TTL. One entry is a CNAME Record for 'zmverify.zoho.com.' with Host 'zb01920222'. Below this is an 'ADD NEW RECORD' button. Further down, there is a section for DNSSEC and Mail Settings. A red arrow points from the 'staticweb' field in the second screenshot to the 'Host' field of the CNAME record.

Once the CNAME record is successfully added, proceed to Azure portal and add the custom domain under the Azure CDN configuration as shown in the image below,

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation bar includes Home, staticwebsitehosting1, swcdn (swcdn/swcdn), Endpoint, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (with sub-options like Origin, Custom domains, Compression, Caching rules, Geo-filtering, Optimization, Rules engine, Locks, Monitoring, Diagnostics logs, Automation, and Tasks (preview)), and Help & support. The main content area displays the 'Add a custom domain' dialog. It shows the 'Endpoint hostname' field containing 'swcdn.azureedge.net'. Below it, the 'Custom hostname' field contains 'staticweb.may...'. A purple rectangular box highlights this 'Custom hostname' field. At the bottom right of the dialog is a blue 'Add' button.

If you proceed to load the website without enabling https you would get security warning error as shown in the image below

Microsoft Azure

Home > staticwebsitehosting1 > swcdn (swcdn/swcdn) Overview

Custom domain Purge Stop Delete

Successfully created custom domain 'staticweb.masterybox.org'

Essentials

Resource group (move) : static-website-demo-rg

Status : Running

Location : Global

Subscription (move) : MSDN Platforms Subscription - \$100 (RN - 10th)

Subscription ID : b7b9d092-6afa-464a-aa2f-019125953c91

Endpoint hostname : http

Origin hostname : http

Protocols : HTTP

Optimization type : General

Custom domains

Hostname	Custom HTTPS	Details
staticweb.mast[REDACTED]	Disabled	

More events in the activity log → Dismiss all

Successfully created a CDN custom domain It can take up to 10 minutes for 'staticweb.mast[REDACTED]' settings to reach every CDN POP. a few seconds ago

Successfully created CDN endpoint Your CDN endpoint 'swcdn' was successfully created. 57 minutes ago

\$97.88 credit remaining Subscription 'MSDN Platforms Subscription - \$100 (RN - 10th)' has a remaining credit of \$97.88. an hour ago

\$77.89 credit remaining Subscription 'my\_main\_subscription\_VSES (RN - 25th)' has a remaining credit of \$77.89. an hour ago

Not Secure https://staticweb.mast[REDACTED]

Soundtrack | Rockstar ... Professional UX UI Des... www.promos.oakwoo... A Sample ISP Compan... Shopping Cart | Konga... Because Of This.... Sto... How to Open and Veri...

Other Bookmarks

Warning: Potential Security Risk Ahead

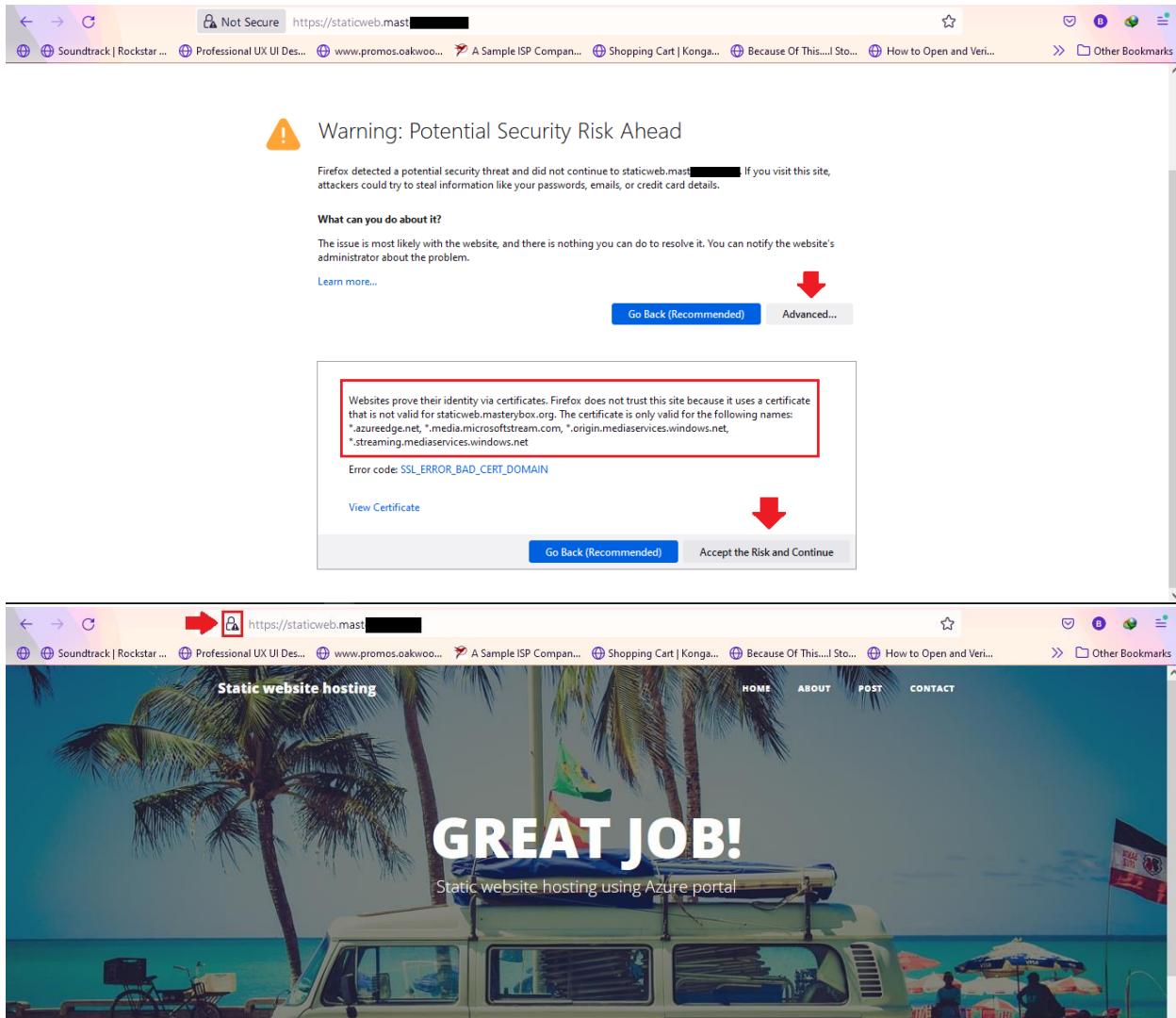
Firefox detected a potential security threat and did not continue to staticweb.mast[REDACTED]. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it. You can notify the website's administrator about the problem.

Learn more...

Go Back (Recommended) Advanced...



## Next Demo:- Static website hosting using Azure DevOps

The next demo we would be hosting a static website using Github, terraform and Azure DevOps pipelines.

Now proceed to enable https

**About This Feature**

Custom Domain HTTPS feature enables you to deliver content to your users securely over your own domain. This is done by encrypting the data between the CDN and your users' clients (typically web browsers) via **TLS protocol** (which is a successor of the SSL protocol) using a certificate. Using our "CDN managed certificate" capability, you can enable this feature with just a few clicks and have Azure CDN completely take care of certificate management tasks such as its renewal. You can also bring your own certificate (stored in [Azure Key Vault](#)) or even purchase a new certificate through Key vault and have Azure CDN use that certificate for securing the content delivery.

[Learn more](#)

**Configure**

Custom domain HTTPS  On  Off

Certificate management type \*  CDN managed  Use my own certificate

Minimum TLS version \*  TLS 1.2  TLS 1.0/1.1

**Status**

- Submitting request  
Your HTTPS request is being submitted.
- Domain validation  
Not started.
- Certificate provisioning

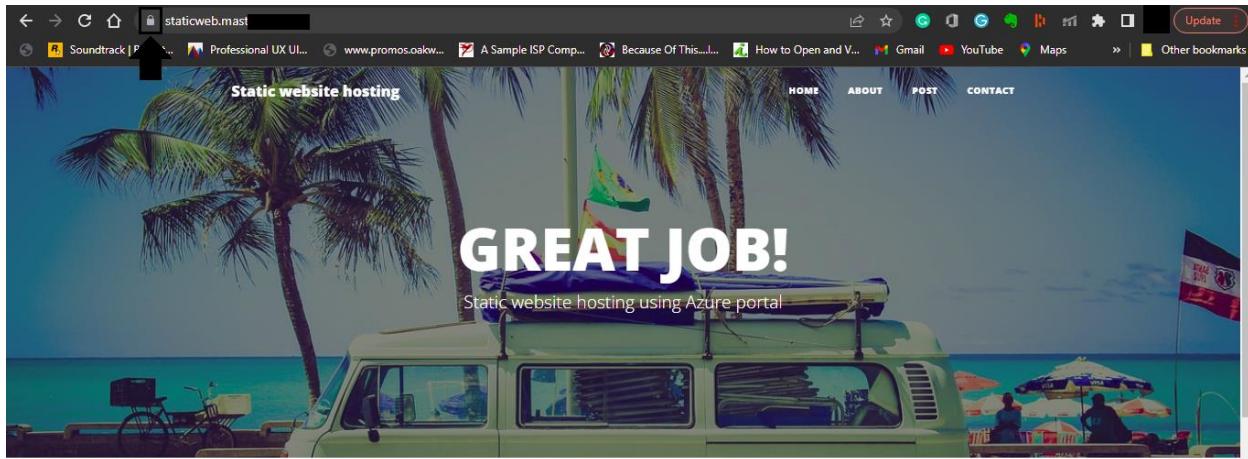
**Notifications**

More events in the activity log → Dismiss all ↻ : all ↻

- Custom domain was successfully updated. Your request is being processed. This custom domain will be automatically validated if a CNAME record exists, mapping this custom domain to the CDN endpoint. If no CNAME record exists, a validation email will be sent to the following email addresses. Please verify the domain as soon as possible. admin@your-domain-name.com, administrator@your-domain-name.com, webmaster@your-domain-name.com, hostmaster@your-domain-name.com, postmaster@your-domain-name.com. a few seconds ago nds ag
- Successfully created a CDN custom domain It can take up to 10 minutes for 'staticweb.master' settings to reach every CDN POP. 17 minutes ago jtes ag
- Successfully created CDN endpoint Your CDN endpoint 'swcdn' was successfully created. an hour ago our ag
- \$97.88 credit remaining Subscription 'MSDN Platforms Subscription - \$100 (RN - 10th)' has a remaining credit of \$97.88. an hour ago our ag
- \$77.89 credit remaining Subscription 'my\_main\_subscription\_VSES (RN - 25th)' has a remaining credit of \$77.89. an hour ago our ag

The screenshot shows two Microsoft Azure portal pages. The top page displays the 'Custom domain HTTPS' configuration for a domain named 'staticweb.mast...'. It shows the 'Custom domain HTTPS' section with 'On' selected for the toggle, 'CDN managed' selected for 'Certificate management type', and 'TLS 1.2' selected for 'Minimum TLS version'. Below this is a 'Status' box with four steps: 1. Submitting request (completed), 2. Domain validation (in progress, indicated by a blue arrow icon), 3. Certificate provisioning (not started), and 4. Complete (not started). The bottom page shows the 'Endpoint' settings for the same static website. It lists the 'Essentials' section with details like Resource group: static-website-demo-rg, Status: Running, Location: Global, Subscription: MSDN Platforms Subscription - 5100 (RN - 10th), and Subscription ID: b7b9d092-6afa-464a-aa2f-019125953c91. The 'Custom domains' section shows a single entry for 'staticweb.mast...' with 'Custom HTTPS' status set to 'Enabling'. Both pages have a large black arrow pointing upwards from the bottom towards the top of the screen.

Now your website is secured proceed to reload the URL.



## **Next Demo:- Static website hosting using Azure DevOps**

The next demo we would be hosting a static website using Github, terraform and Azure DevOps pipelines.



## **RESOURCES**

<https://github.com/balop3e/static-website-demo>

<https://aka.ms/downloadazcopy-v10-windows>

<https://mega.nz/file/YKdBXYIQ#nBIYomzljKF9QXLmdvQwhA45iKLOw-INXyuAJ6uPVU>

## **CONCLUSION**

In this demo we just saw how to host a static website on Azure using the Azure portal. In future posts, I will be hosting this static website using Azure DevOps Pipelines, Git, GitHub and Terraform.