# Create Secure, Authenticated Static Web Site on Azure

**Introduction**

Static web sites are web sites whose contents are not changed for fixed number of days once the content is published. Most of the organizations, small or large, use various static websites for different purposes. Like, a retail organization publishes their monthly catalog to public or an automobile company publishes their car repair manuals to the specific group of people within the organization. These web sites are generally hosted on various servers within the organization for years.

**Problem Statement**

With more and more cloud adaption for most of the companies, their static web sites are also required to be migrated to cloud environment. Couple of weeks back I received one such requirement from one of our esteemed clients who is a giant in automobile industry. They have 100+ static web sites which contain repair manuals which are accessible to thousands of internal users.

Requirement was:

* To create static websites on Azure
* Only authenticated users within the organization can access the site

**Solution Approach**

In this sections I will describe how a secure static web site can be created on Azure which will be accessible only to authenticated users.

Following table describes different solution options considered to create a static website.

|  |  |  |
| --- | --- | --- |
| Solution Option | Pros | Cons |
| Use Azure Storage Account GPv2 native static web site feature | Very easy to configure. A static website can be created in almost no time. | Not secure |
| Host static sites on Azure VM | Conventional approach like current on-premise environment | Cost and management overhead |
| Serverless architecture using Blob Storage and Function Proxy  (Recommended Option) | Secure, no code solution | Take little more time than option 1 to configure. |

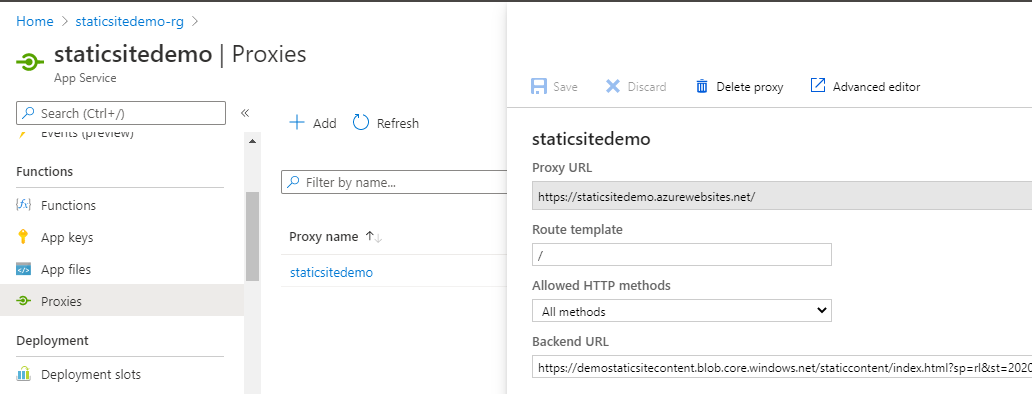
Azure general purpose V2 storage account provides native support for creating static web site. It creates a separate container named $web where the contents need to be uploaded. But, the main problem with this approach is the container must be anonymous. In that case, all the content will be accessible to anyone over internet. Hence, this approach couldn't be taken.

Knowing that default and obvious approach can't be taken, I had to implement an alternative solution. Azure function proxy and a storage account has been used to implement the secure static web site.

First, I created a private container and uploaded static content (HTML, CSS, JS and Images) with their folder structure.

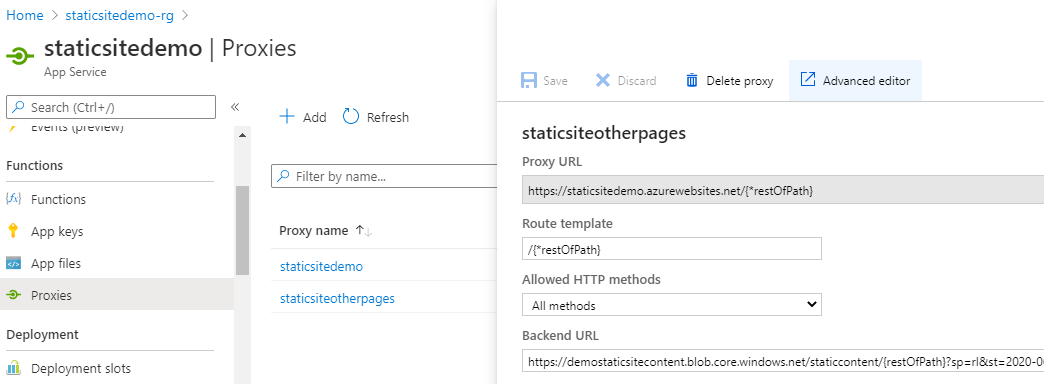
Machine generated alternative text:
demostaticsitecontent I Storage Explorer (preview) 
Storage account 
P Search (Ctrl+/) 
Overview 
Activity log 
PR Access control (IAM) 
Tags 
Diagnose and solve problems 
Data transfer 
Events 
Storage Explorer (preview) 
Searc- 
T • Upload 
Dcnunload Open + New Folder 
staticcontent 
BLOB CONTAINERS 
azure-webjobs-hosts 
azure-webjobs-secrets 
staticcontent 
FILE SHARES 
QUEUES 
TABLES 
Copy URL 
LAST MODIFIED 
e • Select All 
• More 
BLOB TYPE 
CONTENT TYPE 
NAME 
•mages 
bout. html 
contact.html 
index. html 
pricing html 
screenshots.html 
service.html 
team.html 
ACCESS TIER 
Hot (inferred) 
Hot (inferred) 
Hot (inferred) 
Hot (inferred) 
Hot (inferred) 
Hot (inferred) 
Hot (inferred) 
ACCESS TIER LAST MODIFIED 
6/17/2020, AM 
6/17/2020, AM 
6/17/2020, AM 
6/17/2020, AM 
6/17/2020, AM 
6/17/2020, AM 
6/17/2020, AM 
Block Blob 
Block 310b 
Block 310b 
Block 310b 
Block 310b 
Block 310b 
Block Blob 
Folder 
Folder 
Folder 
Folder 
text,'h tm I 
text,'h tm I 
text,'h tm I 
text,'h tm I 
text,'h tm I 
text,'h tm I 
text,'h tm I 
SIZE 
13.5 KB 
10.2 KB 
27.6 KB 
15.3 KB 
12.0 KB 
143 KB 
12.3 KB 
STATUS 
Active 
Active 
Active 
Active 
Active 
Active 
Active 

In order to access the blobs, a function app was created, and a proxy was configured. Backend URL of the proxy was configured with the static site home page (index.html in this demo) blob URL along with container SAS token.



Once created, function proxy will expose a URL. Using this URL blob content can be accessed. At this moment, home page of the static site is accessible via function proxy URL. But the problem is how to access other linked files (html, CSS etc.) of the site as backend URL of the proxy is only pointing to the home page URL.

In order to overcome the above mentioned problem, I had to create one more function proxy. This proxy is configured to accept request for all other pages except the home page.



Main difference between the two function proxies is the "Route template" configuration. "/{\*restOfPath}" is used as route template for the second proxy. Also, backend URL is configured as "*<****blob container URL>/{restOfPath}<container SAS token>***". This will accept all other blob content request from the static site container for the site to work properly.

Final step was to make the site available only for authenticated users. In order to achieve this, authentication/authorization is configured for the function app to use Azure AD authentication.

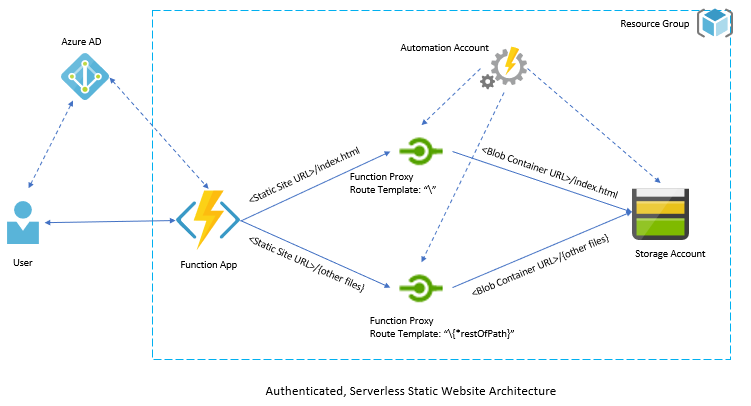
Machine generated alternative text:
staticsitedemo I Authentication / Authorization 
App Service 
P Search (Ctrl+/) 
Deployment 
Deployment slots 
C Deployment Center 
Settings 
Configuration 
Authentication / Authorization 
Application Insights 
Identity 
Backups 
Custom domains 
R] Save X Discard 
To enable Authentication / Authorization, please ensure all y• 
"4.5" or higher and manage pipeline mode is set to Integrat 
App Service Authentication 
Off 
Action to take when request is not authenticated 
Log in with Azure Active Directory 
Authentication Providers 
Azure Active Directory 
Configured (Express Mode Create) 

After configuring this, the site was accessible only to authenticated users.

This way, a secure, authenticated, serverless, cost effective, no code solution can be created to host static web sites.

P.S.: As storage container SAS token, which has an expiry date, has been used in the solution, it requires some operational effort to update the SAS token periodically. This task can be automated easily using Azure automation account and PowerShell runbook.

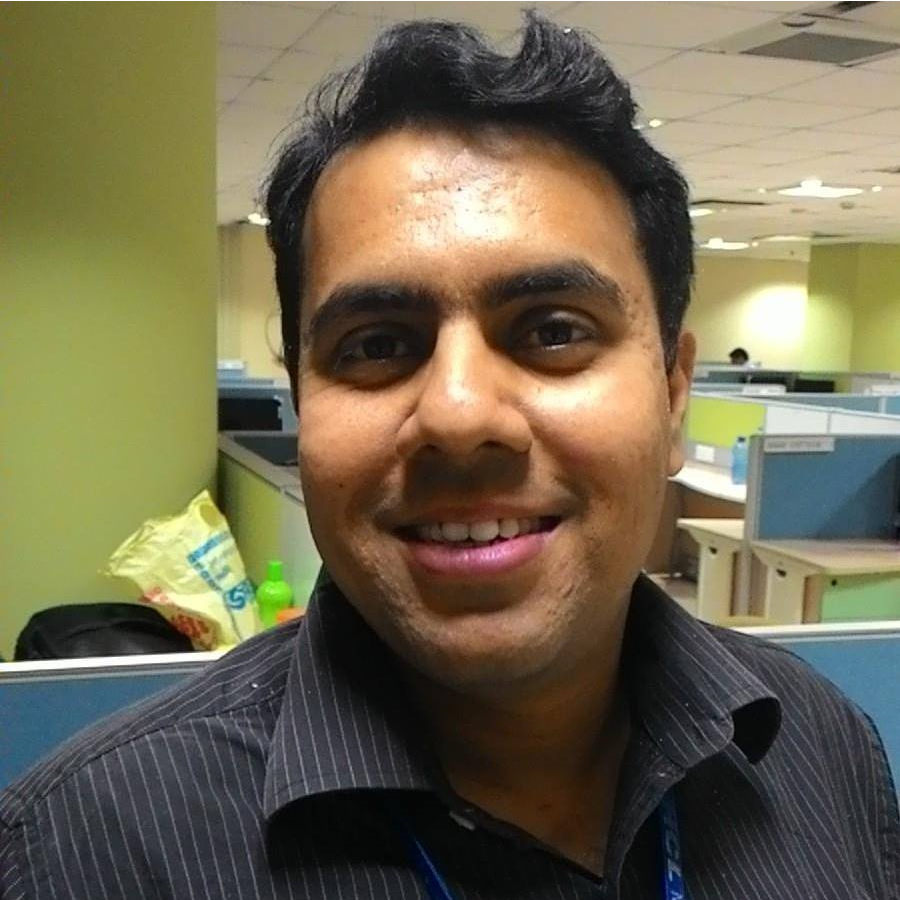
Finally, here is the complete architecture of the provided solution.



**Key Benefits**

* ***Serverless Architecture*** – No effort is required to manage servers. In-built load balancing capability.
* ***No code Solution*** – Static sites can be created with minimum efforts. No code maintenance and upgrade are required
* ***Cost Effective*** – Serverless design with function proxy and blob storage makes it cost effective comparing to IaaS based VM deployment.
* ***Secure*** – Azure AD based authentication ensure only authenticated users can access the site. Even further granular security can be configured so that only specific group of authenticated users will be able to access the site.

**Author**



**Arnab Bhowmick**

**Senior Technical Architect**

[arnab.b@hcl.com](mailto:arnab.b@hcl.com) | (+61) 431945278