Azure scope structure

10 September 2023 20:59

To ease our compliance and access management let's start off by creating two different environment:

- Development / Testing (QA)
- Production (live, user-facing)

We will be organizing this structure using Management Groups.

In real circumstances, Management Groups are used as a way to compartmentalize subscriptions, we end up with a separate billing for each environment.

Governance can be applied at the MG level and inherited to the child resources.

Structure:



We create the Pay-As-You Go subscriptions through the azure portal.

In VS code let's use Azure CLI to push the changes:

```
# global admin azure login
az login
# create parent management group
az account management-group create --name CRC_challenge
# create child management groups
az account management-group create --name Dev_test --parent CRC_challenge
az account management-group create --name Prod --parent CRC_challenge
# list management groups
az account management-group list
# list all subscriptions (first re-auth the CLI with az login!)
az account list
# associate subscriptions to their parent management groups
az account management-group subscription add --name Dev_test --subscription "Subscription 2 - Dev_test"
az account management-group subscription add --name Prod --subscription "Subscription 3 - Prod"
```

Reflected changes:

Name ↑	Туре
🗸 🕅 Tenant Root Group	Management group
Subscription 1 - Root	Subscription
✓ M CRC_challenge	Management group
✓ 🔊 Dev_test	Management group
Subscription 2 - Dev_test	Subscription
V 🕅 Prod	Management group
Subscription 3 - Prod	Subscription

Git: Setting up Version Control

10 September 2023 23:54

We download an html5/css3 template for the website

We first set up source control on Git:

1. Download and install Git Bash, terminal can then be launched within VS code like so

DEBUG CONSOLE	TERMINAL	AZURE		+ ·
MINGW64 ~/Desktop/AZURE/azure-resume-static-website (main)		PowerShell	pw	
		Git Bash	ba	
			Comment Deserves	

- 2. Manually create the resume repo and its readme.md file on Git
- 3. <u>https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent</u> Generate the ssh key for the repository

```
yoan.famel@pc4578945 MINGW64 ~
$ ssh-keygen -t ed8641686 -C "famel.yoan@protonmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/yoan.famel/.ssh/id_ed8641686):
Created directory '/c/Users/yoan.famel/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/yoan.famel/.ssh/id_ed8641686
Your public key has been saved in /c/Users/yoan.famel/.ssh/id_ed8641686.pub
The key fingerprint is:
SHA256:464Ui00P8461HfstF68528ubdcndkcivbb948684 famel.yoan@protonmail.com
The key's randomart image is:
+--[ED25519 256]--+
       ..***++. ..|
       =.0+ ...|
5548 . ..00
        *. ..+.+
        .S E .o *
      ..78. o * .
       +....+
      **=.
               .
     00===
+----[SHA256]----+
```

4. We start the Git ssh-agent and add our private key

```
yoan.famel@pc4578945 MINGW64 ~
$ eval "$(ssh-agent -s)"
Agent pid 678
yoan.famel@pc4578945 MINGW64 ~
$ ssh-add ~/.ssh/id_ed8641686
Enter passphrase for /c/Users/yoan.famel/.ssh/id_ed8641686:
Identity added: /c/Users/yoan.famel/.ssh/id_ed8641686 (famel.yoan@protonmail.com)
```

5. Let's now add the public key to Git

copies the public key to clipboard clip < ~/.ssh/id_ed8641686.pub</pre>

On the web interface we manually add the previously copied public key;

Yoan Famel (yoan-famel) Your personal account				
名 Public profile	Add new SSH Key			
Account				
& Appearance	Title			
骨 Accessibility	ssh-azure-resume-static-website			
D Notifications	Key type			
	Authentication Key 🗢			
🗇 Billing and plans 🛛 🗸	Key			
🖾 Emails	ssh-ed8641686 AAAAC3NzaC1IUHFNhuNfihkn1161154vf46100fknu8FHJoUhdWZD1cu8pUaDF/ze++ fameIyoan@protonmail.com			
Password and authentication				
010 Sessions				
SSH and GPG keys				
Organizations				
Enterprises				
💭 Moderation 🗸 🗸	Add SSH key			

6. Now we clone the remote repo to our local folder

yoan.famel@pc4578945 MINGW64 ~ \$ cd Desktop/AZURE/
yoan.famel@pc4578945 MINGW64 ~/Desktop/AZURE \$ git clone git@github.com:yoan-famel/azure-resume-static-website.git Cloning into 'azure-resume-static-website'…
The authenticity of host 'github.com (180.82.181.3)' can't be established. ED25519 key fingerprint is SHA256:+643513435dmjkmidHdkr4UvCOqU. This key is not known by any other names. Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes
Warning: Permanently added 'github.com' (ED8641686) to the list of known hosts. remote: Enumerating objects: 3, done. remote: Counting objects: 100% (3/3), done. remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 Receiving objects: 100% (3/3), done.

Notice we placed our html5 template in the same root folder. All its content is then copied to a frontend root folder and moved to the local repository.



7. Now we can start reviewing the code structure in VS code, though we will leave the template as is for now and focus on uploading the static website to Azure storage, we will later come back to it and implement the JavaScript counter

yoan.famel@pc4578945 MINGW64 ~/Desktop/AZURE \$ cd azure-resume-static-website/				
<pre>yoan.famel@pc4578945 MINGW64 ~/Desktop/AZURE/azure-resume-static-website (main) \$ code .</pre>				



Let's push the changes to Git before moving on:

stage all the latest changes for our next commit git add -A # creates a snapshot of our current repo state git commit -m "push front end template" # push changes to the remote repo on Git git push

Reflected changes on Git:

ure-resume-static-website / fronte	end / 🖓	Add file 👻 …
Yoan Famel Push the main templa		dc421a9 · 1 hour ago 🕥 History
Name		Last commit date
n		
assets		
images		
LICENSE.txt		
🗋 elements.html		
🗋 generic.html		
🗋 index.html		

Azure storage: Enabling Static Web Hosting

11 September 2023 13:02

1. Let's add the module for Azure Storage to VS code first, that way we can create our storage account without going to the portal

≣ Extension: Azure Storage ×	# main.css • index.html
_	Azure Storage 10.15.3 Prime
	Microsoft @ microsoft.com Ф 1,171,692 ★★★★★(16)
	Manage your Azure Storage accounts including Blob Containers, File Shares, Tables and Queues
	Disable 🔽 Uninstall 🗠 🛞
	This extension is enabled globally.

2. In this case we select the Dev_test subscription we previously created since we are still in the development phase. We will leverage the prod subscription later once our CI/CD pipeline is up and running

■ Extension: Azure Storage	2	Select Subscriptions	3 Selected	ОК
		Subscription 1 - Root 611c8ec3-5b85-4020-8b71-cb9b18fa0a4c		
-		Subscription 2 - Dev_test c3109683-9e25-45b5-9fb5-62a06aec28ba		
	2	Subscription 3 - Prod 8f4b68df-e57d-403a-9349-5c643da9f309		
	-			

3. The Standard_LRS (HDD, locally redundant) storage is created and associated to the dev subscription.

A few basic naming conventions to respect while creating our resources in Azure: <u>https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming</u>



4. By navigating to the blob container we notice that we should first grant RBAC to manage and list its data

 You do not have permissions to list the data using your user account with Azure AD. Click to learn more about authenticating with Azure AD. This request is not authorized to perform this operation using this permission. RequestId:ec094b98-001e-0055-75c2-e490a5000000 Time:2023-09-11T15:16:41.1459731Z

In this case as a global admin I am able to change my permissions. Let's move to the IAM blade of the storage account, this way all child resources inherit the same permissions in case of future blob containers being added to the structure:

Home > All resources > stazureresumestaticweb | Containers > \$web | Access Control (IAM) >

Add role assignment

Role	Members	Conditions (optional) R	eview + assign		
Selected	d role	Storage Blob Data Contribu	utor		
Assign a	access to	Cess to User, group, or service principal Managed identity			
Membe	rs	+ Select members			
		Name	Object ID	Туре	
		Yoan Famel		User	

5. Now, after I made sure no other changes needed to be pushed to Git I moved on to importing the files to Azure storage, but first let's double check it is indeed set as static website:

Home > Resource groups > rgazureres	umestaticweb > stazureresumestaticweb
stazureresumestatio	web∣Static website 🖈 …
₽ Search «	🔚 Save 🗙 Discard 🛛 🖗 Give feedback
Data management	Enabling static websites on the blob service allows you to host static con asynchronously from primary to secondary regions, files at the secondary
🤍 Redundancy	Static website
💡 Data protection	Disabled Enabled
Object replication	An Azure Storage container has been created to host your static website.
Blob inventory	SWED
E Contanualacita	Primary endpoint ①
Static website	https://stazureresumestaticweb.z33.web.core.windows.net/
Lifecycle management	Index document name ①
🙆 Azure search	index.html
Settings	Error document path ①
Configuration	index.html

There are several ways of uploading data to azure storage and I decided to practice the Az-Copy utility with a Shared Access Signature to then realize I had too many problems with this method.

I figured the \$web blob container was the issue and created a test container with a newly generated SAS.

Which worked:



Issues found along the way:

- the SAS token should be placed after the container name and a forward slash
- The \$web container should be escaped with \ since the \$ sign conflicts the command

After trying Az-Copy countless times to no avail I used VS code and imported the frontend folder to the \$web Container that way.



Mapping my custom domain to CDN and enabling TLS

11 September 2023 22:27

1. We can start implementing Azure Content Delivery Network

That way the website can be made accessible to people around the world. It caches and serves content from servers strategically placed worldwide, reducing the distance data has to travel. Making it more available.

This will ensure that my site's resources are quickly available, no matter where the users are located.



After a little while our website can be accessed through Azure CDN



Now that I got the custom domain off of CloudFlare I can associate it to my CDN endpoint.

Let's create a CNAME record so that my source domain yoanfamel.com maps to my destination domain, in this case the CDN endpoint.

All traffic coming from the custom domain will be routed to the endpoint.

I added a few records on CloudFlare,

	Search DNS Records				
	٩			Search	Add record
Туре 🔺	Name	Content	Proxy status	TTL	Actions
CNAME	www	cdn-endpoint-azureresumesta	ticweb.azure 🐔 DNS only	Auto	Edit 🕨
① CNAME	yoanfamel.com	cdn-endpoint-azureresumesta	ticweb.azure 🐔 DNS only	Auto	Edit 🕨

and another for the root. Then I associated the custom domain to the CDN endpoint and enabled HTTPS.

www.yoanfamel.com

Custom domain

🔚 Save 🗙 Discard

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 * ① ODN managed OUse my own certificate

Minimum TLS version * ① • TLS 1.2 TLS 1.0/1.1

We wait for certificate validation:

Status



Now to avoid this scenario:



The account being accessed does not support http.

- HttpStatusCode: 400
- ErrorCode: AccountRequiresHttps
- RequestId : 15e0877f-e01e-004d-1e14-e54fc2000000
- TimeStamp : 2023-09-12T00:56:53.8659512Z

we need to enforce https on users with CDN rules engine:

Select the action to apply to the requests that satisfy the match condition:

- Select Add action, and then select URL redirect.
- For Type, select Found (302).
- For Protocol, select HTTPS.
- Leave all other fields blank to use incoming values.

All we have to do now is refresh the page and observe the redirection when using http.

GitHub Actions: Setting up a CI/CD pipeline

12 September 2023 02:21

Purge CDN endpoint cache on new code Integration

12 September 2023 17:02

Back End API

10 September 2023 23:54

Back End

10 September 2023 23:54