中原大學 雲端計算平台實務

11/26-作業報告

Store data in Azure

資訊四乙 10727211 林彥輝

授課教師：鍾武君 教授

中華民國一一〇年十一月

1. Learning Path Intro

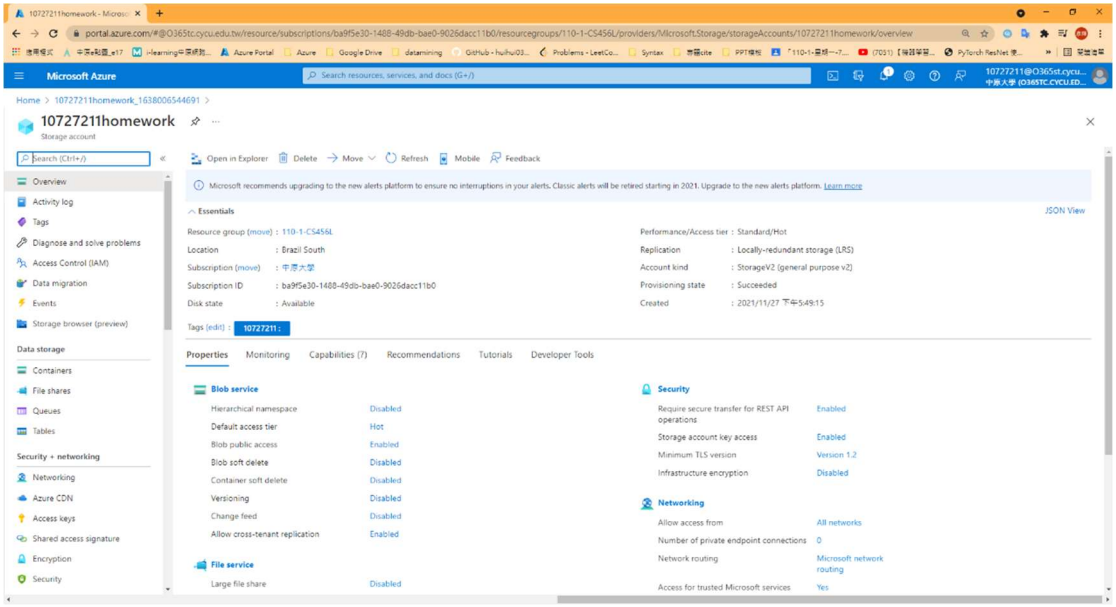
Store data in Azure

https://docs.microsoft.com/en-us/learn/paths/store-data-in-azure/

2. Summary Homework Assignment

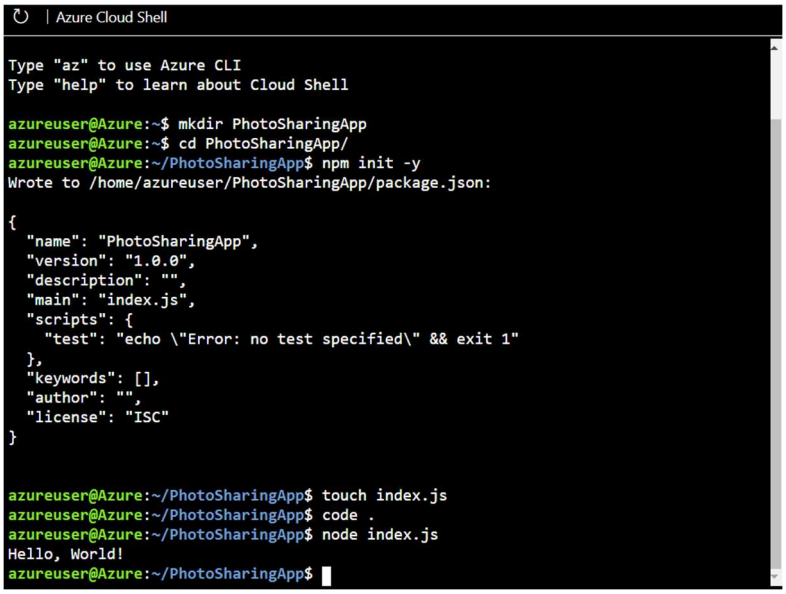
Model 2:Create an Azure Storage account

1. Unit5: Create a storage account using the Azure portal



Model 3:Connect an app to Azure Storage

1. Create a Node.js application



2. Create an Azure storage account

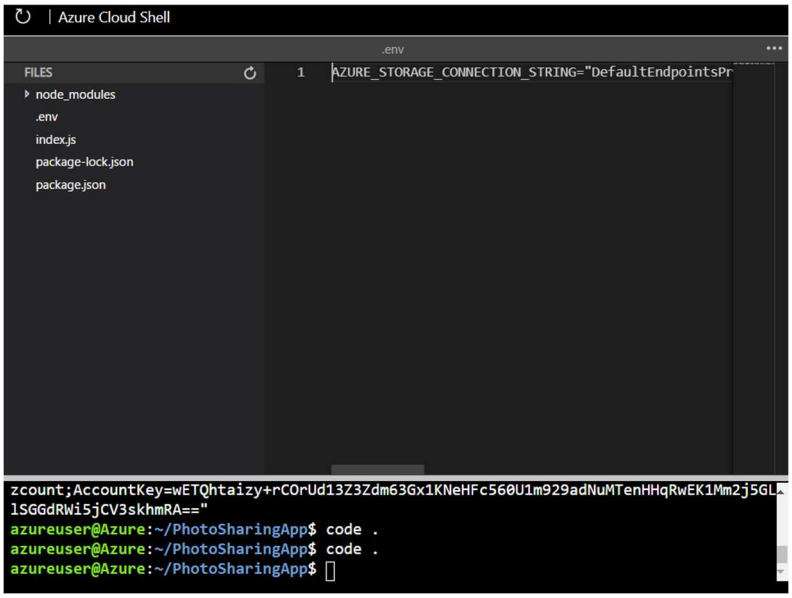
```
azureuser@Azure:~/PhotoSharingApp$ az storage account create \
>   --resource-group learn-4c7467fb-5c33-4e0c-93dd-7158e373a91d \
>   --location westus \
>   --sku Standard_LRS \
>   --name 10727211azcount
{
  "accessTier": "Hot",
  "allowBlobPublicAccess": true,
  "allowCrossTenantReplication": null,
  "allowSharedKeyAccess": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2021-11-26T05:29:12.212226+00:00",
  "customDomain": null,
  "defaultToOAuthAuthentication": null,
  "enableHttpsTrafficOnly": true,
```

3. Add the Azure Storage package

```
azureuser@Azure:~/PhotoSharingApp$ npm install @azure/storage-blob --save
npm WARN PhotoSharingApp@1.0.0 No description
npm WARN PhotoSharingApp@1.0.0 No repository field.

+ @azure/storage-blob@12.8.0
added 36 packages from 135 contributors and audited 36 packages in 9.2s
found 0 vulnerabilities

azureuser@Azure:~/PhotoSharingApp$ node index.js
Hello, World!
azureuser@Azure:~/PhotoSharingApp$
```

4. Create an .env configuration file

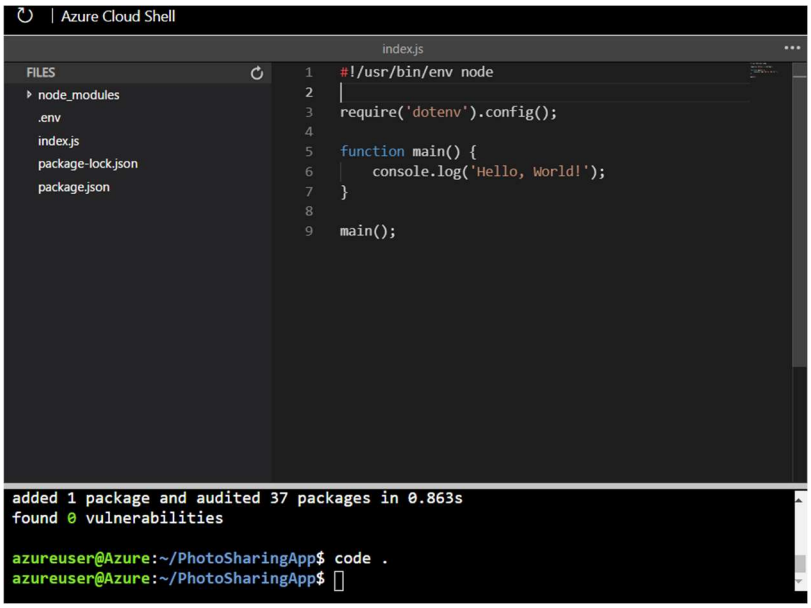
```
azureuser@Azure:~/PhotoSharingApp$ npm install @azure/storage-blob --save
npm WARN PhotoSharingApp@1.0.0 No description
npm WARN PhotoSharingApp@1.0.0 No repository field.

+ @azure/storage-blob@12.8.0
added 36 packages from 135 contributors and audited 36 packages in 9.2s
found 0 vulnerabilities

azureuser@Azure:~/PhotoSharingApp$ node index.js
Hello, World!
azureuser@Azure:~/PhotoSharingApp$ ls
index.js  node_modules  package.json  package-lock.json
azureuser@Azure:~/PhotoSharingApp$ touch .env
azureuser@Azure:~/PhotoSharingApp$ code .
azureuser@Azure:~/PhotoSharingApp$
azureuser@Azure:~/PhotoSharingApp$ az storage account show-connection-string \
>   --resource-group learn-4c7467fb-5c33-4e0c-93dd-7158e373a91d \
>   --query connectionString \
>   --name 10727211azcount
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=10727211a
zcount;AccountKey=wETQhtaizy+rCOrUd13Z3Zdm63Gx1KNeHFc560U1m929adNuMTenHHqRwEK1Mm2j5GL
lSGGdRWi5jCV3skhmRA=="
azureuser@Azure:~/PhotoSharingApp$
```



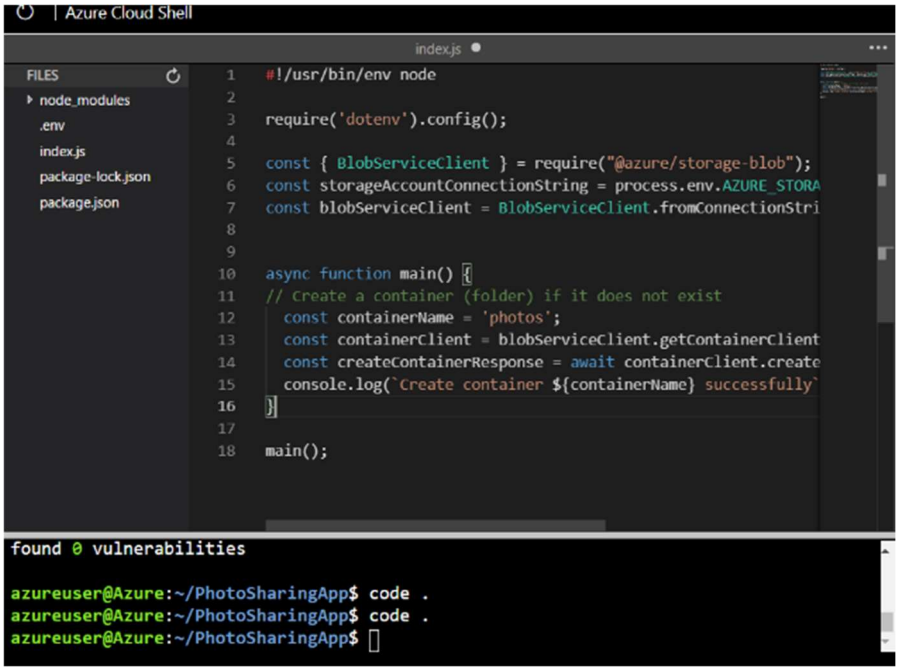
5. Add support to read an environment configuration file

```
azureuser@Azure:~/PhotoSharingApp$ npm install dotenv --save
npm WARN PhotoSharingApp@1.0.0 No description
npm WARN PhotoSharingApp@1.0.0 No repository field.

+ dotenv@10.0.0
added 1 package and audited 37 packages in 0.863s
found 0 vulnerabilities

azureuser@Azure:~/PhotoSharingApp$
```

6. Add code to read the configuration file



7. Azure Storage client library object model

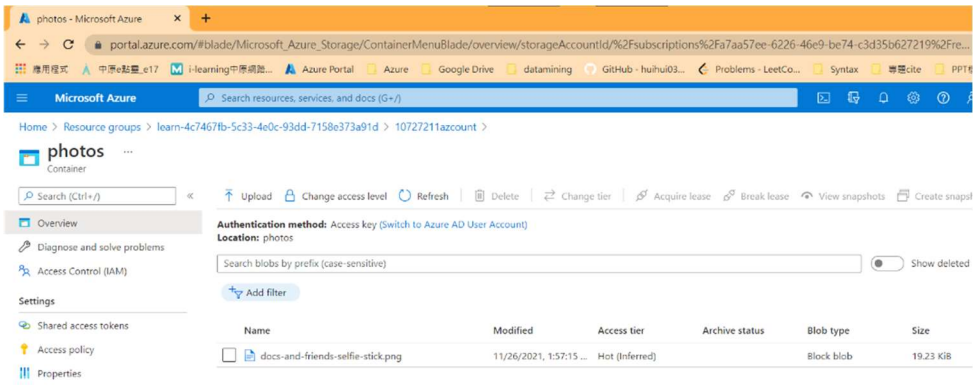



```
azureuser@Azure:~/PhotoSharingApp$ node index.js
Create container photos successfully true
azureuser@Azure:~/PhotoSharingApp$ node index.js
Create container photos successfully false
azureuser@Azure:~/PhotoSharingApp$
```

```
Azure Cloud Shell
azureuser@Azure:~/PhotoSharingApp$ az storage container list \
> --account-name 10727211azcount

There are no credentials provided in your command and environment, we will query for
account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in you
r command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (A
zure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://docs.microsoft.com/az
ure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting cred
entials in your command. Please use --help to get more information about environment
variable usage.
[
  {
    "deleted": null,
    "encryptionScope": {
      "defaultEncryptionScope": "$account-encryption-key",
      "preventEncryptionScopeOverride": false
    },
    "metadata": null,
    "name": "photos",
    "properties": {
      "etag": "\"0x8D9B0A0B1CB067A\"",
      "hasImmutabilityPolicy": false,
```

8. Upload an image to blob storage

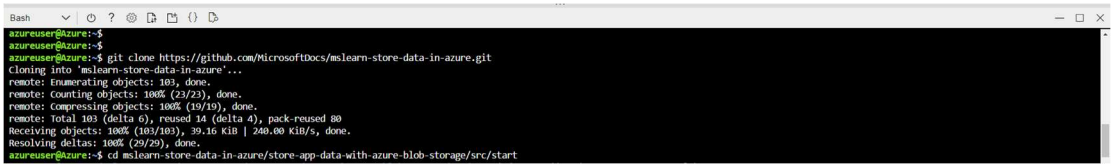
```
Azure Cloud Shell
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/MicrosoftDocs/mslearn-connect-app-to-azure-storage/raw/m
ain/images/docs-and-friends-selfie-stick.png [following]
--2021-11-26 05:53:05--  https://github.com/MicrosoftDocs/mslearn-connect-app-to-azur
e-storage/raw/main/images/docs-and-friends-selfie-stick.png
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/MicrosoftDocs/mslearn-connect-app-to-azur
e-storage/main/images/docs-and-friends-selfie-stick.png [following]
--2021-11-26 05:53:05--  https://raw.githubusercontent.com/MicrosoftDocs/mslearn-conn
ect-app-to-azure-storage/main/images/docs-and-friends-selfie-stick.png
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 1
85.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:
443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19693 (19K) [image/png]
Saving to: 'docs-and-friends-selfie-stick.png'

docs-and-friends-self 100%[===================>]  19.23K  --.-KB/s    in 0s

2021-11-26 05:53:05 (68.6 MB/s) - 'docs-and-friends-selfie-stick.png' saved [19693/19
693]

azureuser@Azure:~/PhotoSharingApp$ code .
azureuser@Azure:~/PhotoSharingApp$ node index.js
Container photos already exists
```



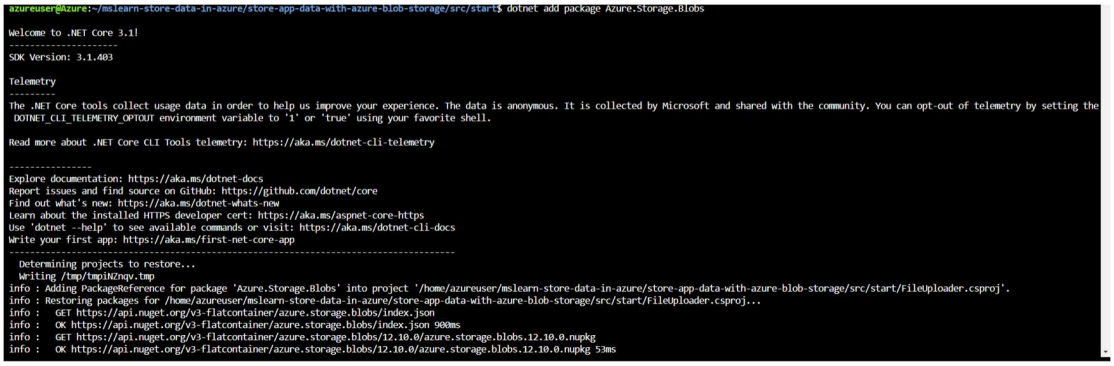
Model 5: Store application data with Azure Blob storage

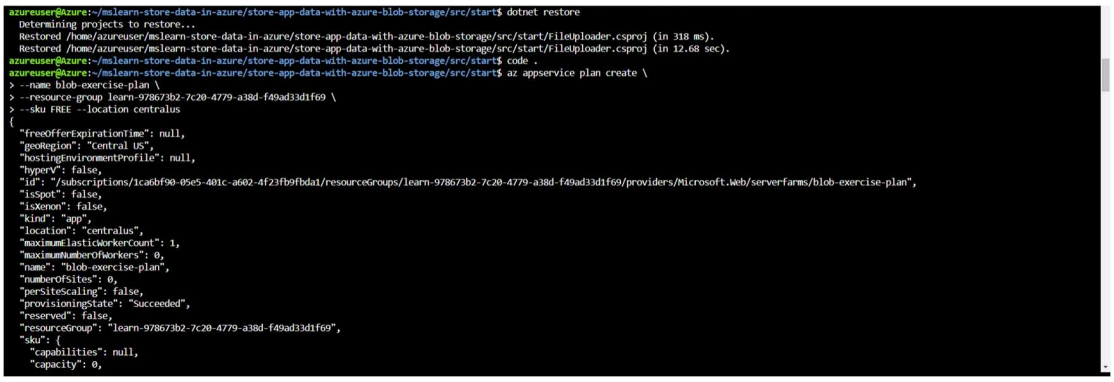
1. Create Storage account

2. Clone and explore the unfinished app



3. Add the NuGet package





4. Models/BlobStorage.cs

BlobStorage.cs

```
using Azure;
using Azure.Storage.Blobs;
using Azure.Storage.Blobs.Models;

using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;
using Microsoft.Extensions.Options;

namespace FileUploader.Models
{
    public class BlobStorage : IStorage
    {
        private readonly AzureStorageConfig storageConfig;

        public BlobStorage(IOptions<AzureStorageConfig> storageConfig)
        {
            this.storageConfig = storageConfig.Value;
        }

        public Task Initialize()
        {
            BlobServiceClient blobServiceClient = new BlobServiceClient(storageConfig.ConnectionString);
            BlobContainerClient containerClient = blobServiceClient.GetBlobContainerClient(storageConfig.FileContainerName);
            return containerClient.CreateIfNotExistsAsync();
        }

        public Task Save(Stream fileStream, string name)
        {
            BlobServiceClient blobServiceClient = new BlobServiceClient(storageConfig.ConnectionString);

            // Get the container (folder) the file will be saved in
            BlobContainerClient containerClient = blobServiceClient.GetBlobContainerClient(storageConfig.FileContainerName);

            // Get the Blob Client used to interact with (including create) the blob
            BlobClient blobClient = containerClient.GetBlobClient(name);

            // Upload the blob
            return blobClient.UploadAsync(fileStream);
        }

        public async Task<IEnumerable<string>> GetNames()
        {
            List<string> names = new List<string>();

            BlobServiceClient blobServiceClient = new BlobServiceClient(storageConfig.ConnectionString);

            // Get the container the blobs are saved in
            BlobContainerClient containerClient = blobServiceClient.GetBlobContainerClient(storageConfig.FileContainerName);

            // This gets the info about the blobs in the container
            AsyncPageable<BlobItem> blobs = containerClient.GetBlobsAsync();

            await foreach (var blob in blobs)
            {
                names.Add(blob.Name);
            }
            return names;
        }

        public Task<Stream> Load(string name)
        {
            BlobServiceClient blobServiceClient = new BlobServiceClient(storageConfig.ConnectionString);

            // Get the container the blobs are saved in
            BlobContainerClient containerClient = blobServiceClient.GetBlobContainerClient(storageConfig.FileContainerName);

            // Get a client to operate on the blob so we can read it.
            BlobClient blobClient = containerClient.GetBlobClient(name);

            return blobClient.OpenReadAsync();
        }
    }
}
```

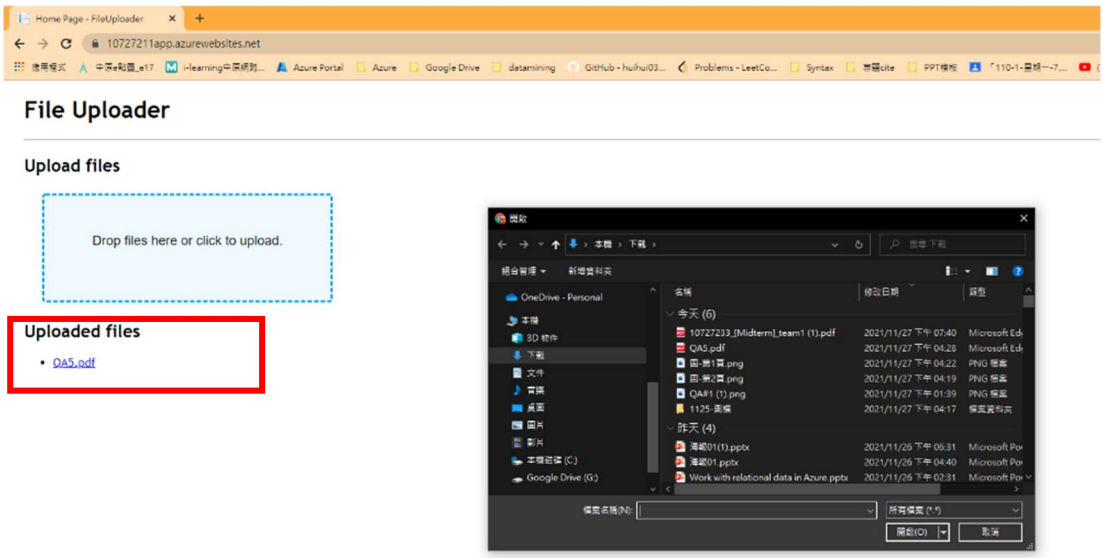
5. Deploy and run in Azure

```
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$ az appservice plan create \
> --name blob-exercise-plan \
> --resource-group learn-978673b2-7c20-4779-a38d-f49ad33d1f69 \
> --sku FREE --location centralus
{
  "freeOfferExpirationTime": null,
  "geoRegion": "Central US",
  "hostingEnvironmentProfile": null,
  "hyperV": false,
  "id": "/subscriptions/1ca6bf90-05e5-401c-a602-4f23fb9fbda1/resourceGroups/learn-978673b2-7c20-4779-a38d-f49ad33d1f69/providers/Microsoft.Web/serverfarms/blob-exercise-plan",
  "isSpot": false,
  "isXenon": false,
  "kind": "app",
  "location": "centralus",
  "maximumElasticWorkerCount": 1,
  "maximumNumberOfWorkers": 0,
  "name": "blob-exercise-plan",
  "numberOfSites": 0,
  "perSiteScaling": false,
  "provisioningState": "Succeeded",
  "reserved": false,
  "resourceGroup": "learn-978673b2-7c20-4779-a38d-f49ad33d1f69",
  "sku": {
    "capabilities": null,
    "capacity": 0,
```

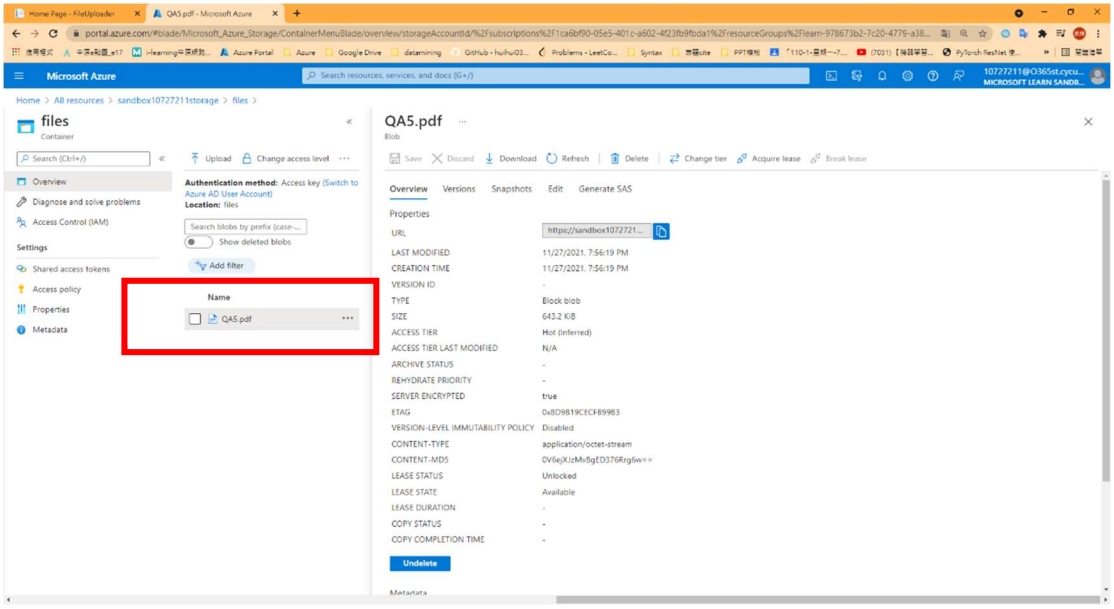
```
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$ az webapp create \
> --name 10727211app \
> --plan blob-exercise-plan \
> --resource-group learn-978673b2-7c20-4779-a38d-f49ad33d1f69
{
  "availabilityState": "Normal",
  "clientAffinityEnabled": true,
  "clientCertEnabled": false,
  "clientCertExclusionPaths": null,
  "clientCertMode": "Required",
  "cloningInfo": null,
  "containerSize": 0,
  "customDomainVerificationId": "CB2AF50B807A9956133F5257A8A91F2857493F4409876365 8B591750A7F998CB",
  "dailyMemoryTimeQuota": 0,
  "defaultHostName": "10727211app.azurewebsites.net",
  "enabled": true,
  "enabledHostNames": [
    "10727211app.azurewebsites.net",
    "10727211app.scm.azurewebsites.net"
  ],
  "ftpPublishingUrl": "ftp://waws-prod-dm1-229.ftp.azurewebsites.windows.net/site/wwwroot",
  "hostNameSslStates": [
    {
      "hostType": "Standard",
      "ipBasedSslResult": null,
      "ipBasedSslState": "NotConfigured",
      "name": "10727211app.azurewebsites.net",
      "sslState": "Disabled",
      "thumbprint": null,
      "toUpdate": null,
      "toUpdateIpBasedSsl": null,
      "virtualIp": null
    },
    {
      "hostType": "Repository",
```

```
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$ CONNECTIONSTRING=$(az storage account show-connection-string \
> --name sandbox10727211storage \
> --output tsv)
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$ az webapp config appsettings set \
> --name 10727211app --resource-group learn-978673b2-7c20-4779-a38d-f49ad33d1f69 \
> --settings AzureStorageConfig:ConnectionString=$CONNECTIONSTRING AzureStorageConfig:FileContainerName=files
[
  {
    "name": "WEBSITE_NODE_DEFAULT_VERSION",
    "slotSetting": false,
    "value": "10.14.1"
  },
  {
    "name": "AzureStorageConfig:ConnectionString",
    "slotSetting": false,
    "value": "DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=sandbox10727211storage;AccountKey=Q8kp39mi5y9Fy20VTnviebSpVCzegPu8Zh7LW0daP2Xk8opg8YehBafcVrgyFaNp2PgNKamSmNIFkxrPXEd8yQ==
",
  },
  {
    "name": "AzureStorageConfig:FileContainerName",
    "slotSetting": false,
    "value": "files"
  }
```

```
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$ dotnet publish -o pub
Microsoft (R) Build Engine version 16.7.0+7fb82e5b2 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

  Determining projects to restore...
  All projects are up-to-date for restore.
  FileUploader -> /home/azureuser/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/bin/Debug/netcoreapp3.1/FileUploader.dll
  FileUploader -> /home/azureuser/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/bin/Debug/netcoreapp3.1/FileUploader.Views.dll
  FileUploader -> /home/azureuser/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/pub/
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start$ cd pub
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/pub$ zip -r ../site.zip *
  adding: appsettings.Development.json (deflated 32%)
  adding: appsettings.json (deflated 23%)
  adding: Azure.Core.dll (deflated 56%)
  adding: Azure.Storage.Blobs.dll (deflated 69%)
  adding: Azure.Storage.Common.dll (deflated 55%)
  adding: FileUploader (deflated 61%)
  adding: FileUploader.deps.json (deflated 91%)
  adding: FileUploader.dll (deflated 55%)
  adding: FileUploader.pdb (deflated 48%)
  adding: FileUploader.runtimeconfig.json (deflated 31%)
  adding: FileUploader.Views.dll (deflated 65%)
  adding: FileUploader.Views.pdb (deflated 49%)
  adding: Microsoft.Bcl.AsyncInterfaces.dll (deflated 45%)
  adding: System.Memory.Data.dll (deflated 43%)
  adding: System.Text.Encodings.Web.dll (deflated 58%)
  adding: web.config (deflated 40%)
  adding: wwwroot/ (stored 0%)
  adding: wwwroot/favicon.ico (deflated 71%)
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/pub$ az webapp deployment source config-zip \
> --src ../site.zip \
> --name 10727211app \
> --resource-group learn-978673b2-7c20-4779-a38d-f49ad33d1f69
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
```

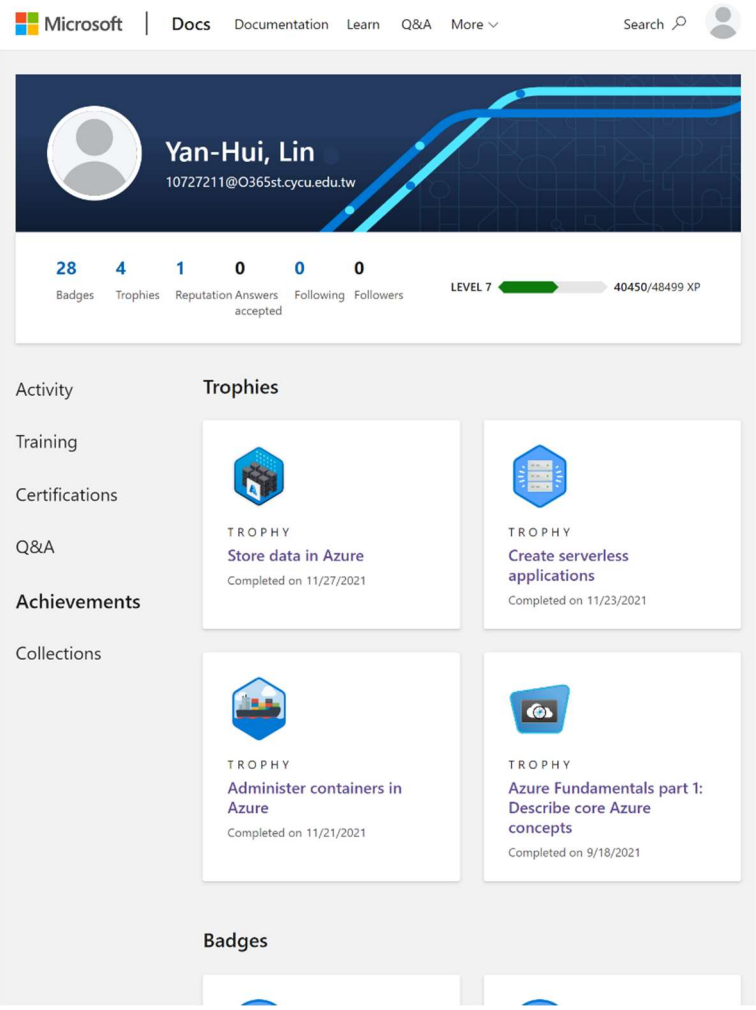

Home Page - FileUploader
10727211app.azurewebsites.net
File Uploader
Upload files
Drop files here or click to upload.
Uploaded files
QA5.pdf

開啟
OneDrive - Personal
Google Drive (G:)
開啟(O)
取消


Home Page - FileUploader
QA5.pdf - Microsoft Azure
Microsoft Azure
files
Container
Authentication method: Access key (Switch to Azure AD User Account)
Location: files
Name
QA5.pdf
QA5.pdf
Blob
Overview
Versions
Snapshots
Edit
Generate SAS
Properties
Undelete


azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/pub$ az storage blob list --account-name sandbox10727211storage --container-name files --query [].{Name:name} --output table
There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.
You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://docs.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.
In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about environment variable usage.
Name
QA5.pdf
azureuser@Azure:~/mslearn-store-data-in-azure/store-app-data-with-azure-blob-storage/src/start/pub$

Take screenshots of Badges and Trophies



Learned from the Learning Path

在 Model1 簡單介紹資料格式儲存的種類，透過種類更清楚各種 Azure Storage 的適當使用時機。Model2-3 開始實作如何創建 Storage Account，並且透過簡單的程式透過 Storage 的金鑰與 Storage 做互動，在 Model4 中了解 Azure 背後的安全性機制，在最後的 Model5 簡單實作一隻程式能與 Blob 作互動。

Problems

整體實作相當流暢，並無遇到任何困難。但在課堂上有碰到其他人在實作的過程中有困難，原因主要為指令輸入錯誤（每個人在 sandbox 的 resource group 不同，而使用投影片的指令將會使用他人的 resource group，導致 error）。

FeedBack

整體的教學解說精細、範例程式都能正常 work，是個相當適合新手的教學，但仍希望在中文版翻譯中，一些名詞能在後面補充原文，例如佇列 (Queue)。