```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "d2ab2199",
   "metadata": {},
   "source": [
    "# Lab3: Cloud Storage Services and Setup\n",
    "\n",
    "Maryam R.Aliabadi, August 7th, 2023"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "16c7018f",
   "metadata": {},
   "source": [
    "## Learning objectives\n",
    "\n",
    "The goal of this lab is to launch and configure an Azure Blob Storage for use within your team. You will learn:\n",
    "\n",
    "- Setting up an Azure Blob Storage\n",
    "- Moving data into your Blob Storage\n",
    "- Reading data from your Blob Storage"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a2621106",
   "metadata": {},
   "source": [
    "## Storage\n",
    "\n",
    "- Managed Disks - Azure Managed Disks\n",
    "- Azure Files - Azure Files\n",
    "- Blob Storage - Azure Blob Storage\n",
    "\n",
    "### Managed Disks\n",
    "\n",
    "Azure Managed Disks are block storage, which means the disk utilization happens in blocks or data stored in blocks. A ***file is stored in blocks***, so if we want to change one character in a one GB file, we just want to change the block that contains that one bit. This is how data is stored, and it functions very similar to your laptop hard disk, flash drive, or any external disk. We use this as storage for our Azure Virtual Machine.\n",
```

```
    "\n",
    "So we can use Managed Disks as a boot volume or attach it to an existing Azure Virtual Machine, just like how we connect an external hard disk to our laptop.\n",
    "\n",
    "```{note}\n",
    "Azure Virtual Machines need an Azure Managed Disk to boot.\n",
    "```\n",
    "\n",
    "Here are some of the properties of Managed Disks:\n",
    "\n",
    "- Persistent storage (non-volatile storage)\n",
    "- Automatically replicated within the availability zones \n",
    "- High availability and durability\n",
    "- More durability by backing up (taking snapshots) data to Blob Storage. \n",
    "- Low latency\n",
    "- Scale storage up or down\n",
    "- Data encryption \n",
    "\n",
    "### Azure Files\n",
    "Provides storage for Azure Virtual Machines; unlike Managed Disks, Azure Files storage can be accessed by multiple Azure Virtual Machines simultaneously. Azure Files offers all the advantages that we mentioned with Managed Disks but also provides the following on top of it:\n",
    "\n",
    "- Auto-scaling \n",
    "- Replication on multiple availability zones\n",
    "- Sharing with other Azure Virtual Machines\n",
    "\n",
    "This is mainly used in industry to provide home directories for workers.\n",
    "\n",
    "```\n",
    "Azure Files won't be replacing Managed Disks; both have their use cases. For example, we need Managed Disks storage for an Azure Virtual Machine as a boot drive. But, Azure Files, you can think more in a corporate environment performing centralized shared storage—uses like media processing or shared code repositories.\n",
    "```\n",
    "\n",

"[Here](https://learn.microsoft.com/en-us/azure/architecture/aws-professional/storage) is an article that shows the difference between different storage solutions Azure provides and their equivalent services in AWS.\n",
    "\n",
```

```
    "### Blob Storage\n",
    "\n",
    "Blob Storage is an object-level storage, where a ***file is stored as an entire object***, so if we want to change one character in a one GB file, we want to update the file, and then we have to replace that entire file. By the object storage, you can think of it as the ***Key-Value store***, where the key is the filename, and the Value is the contents of the object or the file itself. So you store an object with a Key and retrieve the object with the key. \n",
    "\n",
    "Summarizing some properties of Blob Storage:\n",
    "\n",
    "- Data is stored as objects in containers\n",
    "- Virtually unlimited storage (Single object is limited to 5 TB)\n",
    "- Designed for 99.999999999% of durability\n",
    "- Granular access to containers and objects\n",
    "\n",
    "You don't need to specify the availability zones, as Azure will take care of all these in replicating your data across different availability zones."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "087dec40",
   "metadata": {},
   "source": [
    "## Setting up Azure Blob Storage\n",
    "\n",
    "[Setup your Azure Blob Storage](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction) \n",
    "\n",
    "Before attempting to read or upload to your Azure Blob Storage, you should ensure that the storage account and container exist and you have access to it. \n",
    "\n",
    "Click the toggle below for instructions to create and setup your storage account and container:\n",
    "\n",
    "- Go to the Azure portal.\n",
    "- Search storage accounts, navigate and click create.\n",
    "- Select the subscription and resource group.\n",
    "- Provide a name for the storage account (e.g.:`mds-blob-14-gittu`).\n",
    "- Make sure Azure region is `(US) West US`.\n",
```

```
    "- Go to Advanced setting, select \"Allow enabling anonymous access on individual containers\"\n",
    "- Next in Networking section, when creating the storage account, make sure you select \"Enable public access from all networks\"\n",
    "- All other options leave as it is. And click on `Create`.\n",
    "- After your storage account is created, create a container in this storage account.\n",
    "    - Click on your storage account\n",
    "    - Click on Containers under Data Storage\n",
    "    - Click on `+ Container` button\n",
    "    - Provide a name for your new container (e.g.: `mds-container-14-gittu`)\n",
    "    - Select an option for the Public access level.\n",
    "    - Now click on the `Create` button to create the container.\n",
    "\n",
    "Please note that making a storage account or container public so that anyone, including your team members, can access it is generally not recommended due to security concerns. It's better to manage access at a more granular level using Azure's built-in security features.\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "0b804540",
   "metadata": {},
   "source": [
    "### More about credentials\n",
    "\n",
    "Azure provides a simple way to share blob storage as well set policies to control access to the blob storage. We cam generate a shared access signature (SAS) for the blob storage, which is a URI that grants restricted access rights to the storage account. The SAS can be used to access blob storage without requiring the user to have an Azure account or to know the storage account key. For a more secure way, we can use Azure Active Directory (Azure AD) to authenticate a user to the storage account and provide access. However, azure account also provides creating policies for each container, which is a more granular way of controlling access to the blob storage. We can create a policy, generate the SAS token, and share it with the team members. Once the work is done, we can revoke the access by deleting the policy.\n",
    "\n",
    "### Accessing Azure Blob Storage\n",
    "1. Go to the blob container that you created\n",
    "2. Click on the `Access Policy` button\n",
    "3. Add a policy, provide a name, permissions, and expiry date. Once done, click 'Save'\n",
```

```
    "4. Now navigate to the `Shared access tokens` page\n",
    "5. You can now choose the policy you created to generate the SAS token
and URL\n",
    "6. Copy and share the SAS token and URL with your team members to get
access to the blob storage "
   ]
  },
  {
   "cell_type": "markdown",
   "id": "c4550f7d",
   "metadata": {},
   "source": [
    "## How to transfer data to Azure Blob Storage\n",
    "\n",
    "There are many ways you can put data into Azure Blob Storage. You can
do it using...\n",
    "\n",
    "### Web interface \n",
    "\n",
    "You can upload files using the Azure portal.\n",
    "\n",
    "### SDK \n",
    "\n",
    "You can use the Azure SDKs offered in multiple languages.\n",
    "\n",
    "### Using CLI\n",
    "You can use the Azure CLI and AzCopy to upload data. If you want to
upload a folder or a file, AzCopy is a good choice.\n",
    "If you are logged in to Azure CLI, you can use the following command
to upload a file to Blob storage:\n",
    "```bash\n",
    "azcopy copy \"<local-folder-path>\"
\"https://<storage-account-name>.blob.core.windows.net/<container-name>\"
--recursive=true\n",
    "```\n",
    "\n",
    "Here is an example command to upload files using the SAS token incase
your account does not have access to the storage account:\n",
    "\n",
    "```bash\n",
    "azcopy copy \"<local-folder-path>\"
\"https://<storage-account-name>.blob.core.windows.net/<container-name>?<SA
S token>\" --recursive=true\n",
    "```\n",
    "\n",
    "Please replace `<local-folder-path>`, `<storage-account-name>`,
```

```
`<container-name>`, `<SAS Token>` with your local folder path, your storage account name, your container name, and your SAS token respectively.\n",
    "\n",
    "The --recursive=true option ensures that all files in the directory and its subdirectories are uploaded.This command is similar to the cp -R command in Linux and the aws s3 cp command in AWS CLI, but it’s for Azure Blob Storage.\n",
    "\n",
    "You can also refer to [this](https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10) document for more details on using the Azure CLI with Azure Storage."
   ]
  },
  {
   "cell_type": "markdown",
   "id": "516af748",
   "metadata": {},
   "source": [
    "## How to read data from Azure Blob Storage\n",
    "\n",
    "Files can directly be read from Azure Blob Storage using the shared credentials. If you hit the URL `https://<storage-account-name>.blob.core.windows.net/<container-name>/<file_name>?<SAS token>`, you can view your file in the browser.\n",
    "Popular data processing libraries like pandas can directly access it using the SAS url you provided above.\n",
    "\n",
    "Moreover, there are libraries from azure as `azure-storage-blob` which provides clients that can access Azure Blob Storage directly using the shared credentials.  \n",
    "\n",
    "To start, let's install the necessary packages using pip. Then, we can easily read files from Azure Blob Storage and start analyzing the data.\n",
    "\n",
    "Here is an example of how you can access Azure Blob Storage using the shared credentials:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "019cc0f8",
   "metadata": {},
   "outputs": [],
   "source": [
```

```
    "import os\n",
    "from azure.storage.blob import BlobServiceClient\n",
    "import pandas as pd\n",
    "\n",
    "# Your SAS token\n",
    "SAS_token = \"<SAS token>\"\n",
    "\n",
    "# Storage account and container names\n",
    "STORAGEACCOUNTNAME = \"<storageaccountname>\"\n",
    "CONTAINERNAME = \"<containername>\"\n",
    "\n",
    "# Create a blob service client with SAS token\n",
    "blob_service_client = BlobServiceClient(account_url=\"https://\" +
STORAGEACCOUNTNAME + \".blob.core.windows.net\", credential=SAS_token)\n",
    "\n",
    "# Get a reference to the container\n",
    "container_client =
blob_service_client.get_container_client(CONTAINERNAME)\n",
    "\n",
    "# List all blobs in the container\n",
    "blob_list = container_client.list_blobs()\n",
    "for blob in blob_list:\n",
    "    print(blob.name)"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "15464ce2",
   "metadata": {},
   "source": [
    "Please replace `<STORAGEACCOUNTNAME>`, `<SAS_token>`, and
`<CONTAINERNAME>` with your specific values.\n",
    "If you want to see how to install necessary packages in TLJH, please
check above section on How to install packages in TLJH."
   ]
  },
  {
   "cell_type": "raw",
   "id": "17c38c41",
   "metadata": {},
   "source": [
    "sudo -E pip install azure-storage-blob\n",
    "sudo -E pip install pandas"
   ]
  },
  {
```

```
   "cell_type": "code",
   "execution_count": null,
   "id": "211a878b",
   "metadata": {},
   "outputs": [],
   "source": [
    "import json\n",
    "import urllib.parse\n",
    "import pandas as pd"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "7bef9264",
   "metadata": {
    "tags": []
   },
   "outputs": [],
   "source": [
    "%%time\n",
    "df =
pd.read_parquet(\"https://<storage-account-name>.blob.core.windows.net/<con
tainer-name>/<file_name>?<SAS token>\", \n",
    "        filters=[('year','=', 2004)], \n",
    "                columns=['year', 'UniqueCarrier', 'ArrDelay'])\n",
    "print(df[(df.year== 2004) & (df.ArrDelay >
10)][\"UniqueCarrier\"].value_counts())\n",
    "\n",
    "#Or alternatively\n",
    "\n",
    "# Get a reference to the blob\n",
    "blob_client = container_client.get_blob_client(\"<blob-name>\")\n",
    "# Download the blob to a string\n",
    "data = blob_client.download_blob().readall().decode('utf-8')\n",
    "df = pd.read_parquet(data, filters=[('year','=', 2004)],
columns=['year', 'UniqueCarrier', 'ArrDelay'])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "bac4713f",
   "metadata": {
    "tags": [
     "skip-execution"
```

```
    ]
   },
   "outputs": [],
   "source": [
    "## just getting 1000 rows to save time\n",
    "df = pd.read_csv(data,nrows = 1000)\n",
    "print(df[(df.year== 1996) & (df.ArrDelay >
10)][\"UniqueCarrier\"].value_counts())"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "c4a02dc9",
   "metadata": {},
   "source": [
    "## Azure Archive Blob Storage (optional)\n",
    "\n",
    "Azure Archive Blob Storage is an extremely low-cost cloud service for
long-term backup (data archiving service). The drawback is that it takes
several hours to retrieve the stored data and should only be used for
archiving. The retrieval time depends on the retrieval option that the user
goes for. There are mainly 3 retrieval options...\n",
    "\n",
    "- Bulk (5 - 12 hours)\n",
    "- Standard (3- 5 hours)\n",
    "- Expedited (1 - 5 minutes)\n",
    "\n",
    "Cost increases as the speed of retrieval increases.\n",
    "\n",
    "There are 3 main elements of Azure Archive Blob Storage...\n",
    "\n",
    "- Blob\n",
    "All files (Any object such as a photo, video file, or document) should
be zipped before uploading to the Azure Archive Blob Storage. This is
considered the base unit of storage. All have a unique ID and also have a
description.\n",
    "- Container\n",
    "It is a container for storing blobs, and you can specify the region
where you want to locate your container.\n",
    "- Access policies\n",
    "This is how you control access to the container. You can give
permissions to individuals and what kind of operation they can do.\n",
    "\n",
    "Can you think of use-cases for storing data in Azure Archive Blob
Storage?\n",
    "\n",
```

```
    "In industries, data usually follows a lifecycle, and we can achieve this life cycle using Azure Blob Storage and Azure Archive Blob Storage.\n",
    "\n",
    "[Here](https://azure.microsoft.com/en-us/products/storage/) you can read some best practices with Azure Blob Storage."
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.11.1"
  },
  "vscode": {
   "interpreter": {
    "hash": "6e9e0baa62560f8a3b402c12d339bdad33c58a25305700ec7e7682c0b6251f68"
   }
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```