This is a mini project using Azue virtual machine, and Azure storage blob container. The task is to download page 50 of the API web page at <https://www.themuse.com/developers/api/v2>

[**https://www.themuse.com/api/public/jobs?page=50**](https://www.themuse.com/api/public/jobs?page=50)

, then obtain the following data:  publication date, job name, job type, job location, company name

which can be found on the “Response body”

Below are the steps in detail

**1/ Create Ubuntu virtual environment on the VM** including installing all of the required tools/libraries, such as azcopy, etc

python3 -m venv venv

source venv/bin/activate

pip install requests

pip install pandas

pip install toml

sudo apt update

sudo apt install snapd

sudo snap install azcli #this will install azcopy as well

azcopy –version # verify if azcopy was installed successfully

**2/ Create a virtual machine on Azure**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

setting connection to VM from VS code

A screenshot of a computer

Description automatically generated

A computer code with black text

Description automatically generated

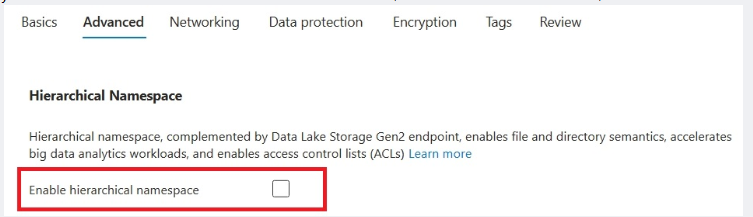
**3/ Create a storage account on Azure, and blob container**

A screenshot of a computer

Description automatically generated

A close-up of a text

Description automatically generated



A screenshot of a computer

Description automatically generated

After creating storage account, we then create blob container, and create shared access token on the container to generate url for accessing using subprocess in the python script

A screenshot of a computer

Description automatically generated

**4/ write python script** to download data from the above API webpage, transform, and then upload it to the blob container. Please see below python script (note: config.toml is not disclosed due to confidentiality)

*#down load data from the API*

def read\_data(url: str) -> list[dict]:

    try:

        response = requests.get(url)

        if response.status\_code == 200:

*#only get the results*

            data = response.json()['results']

            if isinstance(data,list):

                return data

            else:

                print('the data received is not a list of dictionary')

        else:

            print('failed to load data from the API. Status code is {response.status\_code}')

    except Exception as e:

        print(' an error occured {e}')

*#extract downloaded data*

def extract\_data(data: list[dict]) -> list[list[str], list[str], list[str], list[str], list[str]]:

    companies = [item['company']['name'] for item in data]

    locations = [item['locations'][0]['name'] for item in data]

    jobs = [item['name'] for item in data]

    job\_type = [item['type'] for item in data]

    publication\_date = [item['publication\_date'] for item in data]

    return companies, locations, jobs, job\_type, publication\_date

if \_\_name\_\_ == '\_\_main\_\_':

    import requests

    import pandas as pd

    import subprocess

    from typing import List

    import toml

    config\_data = toml.load('config.toml')

    url = config\_data['config']['url']

    shared\_key = config\_data['config']['shared\_key']

    data = read\_data(url)

    li = extract\_data(data)

*# convert the data (list of list) by transposing first, then convert to data frame, and lastly export to filename*

    transposed\_list = list(map(list,zip(\*li)))

    df = pd.DataFrame(transposed\_list, columns=['Company','Location','Job','Job\_type','Publication\_date'])

    df[['City','Country-State']]=df['Location'].str.split(',',expand=True)

    df = df.drop(['Location'],axis=1)

    df.to\_csv('demo.csv',index=False)

*#upload file to blob container on azure storage account*

*#recursive to allow copy multiple files just in case*

    subprocess.run(['azcopy','copy','demo.csv', shared\_key,'--recursive=true','--overwrite=true'])

**5/ Verify if the upload was done successfully**

A screenshot of a computer

Description automatically generated